Republic of Iraq

Ministry of Higher Education & Scientific Research

University of Kerbala

College of Engineering

Department of Electrical and Electronic Engineering

Automatic Traffic Rules Violation Detection System

A Thesis Submitted to the Council of the College of Engineering/University of Kerbala in a Partial Fulfillment of the Requirements for the Degree of Master of Science (M.Sc.) in Electrical Engineering

Written By:

Ali Qasim Abd Ali

Supervised By:

Assist. Prof. Dr. Hameed R. Farhan

Assist. Prof. Dr. Muayad S. Kod

June 2024 Dhu al-Hijjah 1445

بسم الله الرحمن الرحيم

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ

صدق الله العلي العظيم

(المجادلة: من الآية 11)

Examination committee certification

We certify that we have read the thesis entitled "**Automatic Traffic Rules Violation Detection System**" and as an examining committee, we examined the student "**Ali Qasim Abd Ali**" in its content and in what is connected with it and that, in our opinion, it is adequate as a thesis for the degree of Master of Science (M.Sc.) in Electrical Engineering.

Supervisor

Signature:
Name: Assist. Prof. Dr.Hameed R. Farhan
Date: 4/ 8/ 2024

Supervisor

Signature:
Name: Assist. Prof. Dr. Muayad S. Kod
Date: 5 / 8 / 2024

Member

Signature: Jabbar S. Hussein
Name: Assist. Prof. Dr. Jabbar S. Hussain
Date: 4/ 8/ 2024

Member

Signature:
Name: Dr. Arwa H. Mohammed
Date: 3 /8/ 2024

Chairman

Signature:
Name: Prof. Dr. Haider Galil Kamil
Date: 4/ 8/ 2024

Signature:
Name: Assist. Prof. Dr. Muayad S. Kod
Head of the Department of Electrical and Electronic Engineering
Date: 7 / 8/ 2024

Signature:
Name: Prof. Dr. Haider Nadhom Azziz
Dean of Engineering College
Date: 6 / 8/ 2024

Supervisor certificate

We certify that the thesis entitled **"Automatic Traffic Rules Violation Detection System"** was prepared by **Ali Qasim Abd Ali** under our supervision at the Department of Electrical Engineering, College of Engineering, University of Kerbala as a partial fulfilment of the requirements for the Degree of Master of Science (M.Sc.) in Electrical Engineering.

Signature:

Assist. Prof. Dr. Hameed R. Farhan

Date: / / 2024

Signature:

Assist. Prof. Dr. Muayad S. Kod

Date: / / 2024

Linguistic certificate

I certify that the thesis entitled "**Automatic Traffic Rules Violation Detection System** " which has been submitted by **Ali Qasim Abd Ali** has been proofread and its language is amended to meet the English style.

Signature: Hussein

Dr. Hussein Ali Hadi

Date: 5 / 8 / 2024

Undertaking

I certify that the research work "**Automatic Traffic Rules Violation Detection System**" is mine. The work has not been presented elsewhere for assessment, and the materials used from other sources have been properly acknowledged/referred.

Signature:

Ali Qasim Abed Ali

Date: 7/8/ 2024

Dedication

To my beloved Lord, the owner of the era and time, whose presence in life deserves all this effort, steadfastness, and useful knowledge

To the Lord on whom we place our hopes so that we may see from Him security with which to continue our lives

To straighten my back, my dear father

To whom does the tranquility of prostration resemble, whose feet are Paradise, my compassionate mother

To my beloved wife, Sanar, and my children, Fadl, Haider, and Mahdi

To my dear sisters Shaima and Zainab.

Acknowledgements

All thanks, praise and gratitude to Allah Almighty for his kindness and benevolence to us as he brought us out of darkness into light... Out of gratitude, I am pleased to extend my great thanks and gratitude to the two professors who supervised my thesis (Assist. Prof. Dr. Hameed R. Farhan and Assist. Prof. Dr. Muayad S. Kod), who have knowledge and high morals, for their advice and sound guidance since the early stages of this work and without boredom or fatigue, as they shared their deep experiences in all fields of research, and may Allah prolong their lives as a beacon in the path of science. Finally, I would like to thank everyone who extended a helping hand and supported me and stood beside me in obtaining information, guidance or advice that enabled me to walk a steady pace in my scientific career, asking Allah to reward them with the fullest reward.

Abstract

Traffic violations cause significant problems such as congestion, accidents, and deaths. It is highly desirable to have an effective automated system to detect and record these violations, thus, to improve traffic regulation enforcement and reduce human intervention. This study aims to detect traffic violations and identify violated vehicle plates in Iraq. As the system detects traffic violations automatically, of three different cases; non-compliance with the traffic signal, exceeding the speed limit, and driving in the opposite direction, it is cost-effective, practical, and powerful. The proposed system uses background subtraction technology to detect moving vehicles and the concept of time and distance difference over which vehicles move to detect violations. The You Only Look Once version 4 (YOLOv4) algorithm is used to identify the license plates (LPs) of violating vehicles with great accuracy. The Convolutional Recurrent Neural Networks (CRNN) were used with Optical Character Recognition (OCR) technology to recognize two types of Iraqi LPs; the first type contains Hindi numbers, while the second form includes Arabic numbers, a future system to be used throughout Iraq. The results achieved from our trial indicate promising system performance, with multiple real-time violation detection and an overall accuracy rate of 98.06% for offenses. On the other hand, the proposed system has excelled in recognizing LPs, achieving a success rate of 94.87% and 98.22% for the first and second types, respectively. It has outperformed similar systems, particularly in terms of accuracy and the capability to detect multiple types of traffic violations simultaneously, thereby establishing its competitive edge.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ALPR	Automatic License Plate Recognition
BPNN	BackPropagation Neural Network
BLOB	Binary Large Object
CCPD	Chinese City Parking Dataset
CTC	Connectionist Temporal Classification
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
GMM	Gaussian Mixture Model
GAC	Gradient Angle Changes
GBDT	Gradient-Boosted Decision Tree
HSA	Hough Space Analysis
IoU	Inter Section Over Union
IoT	Internet of Things
ILPR	Iraqi License Plate Recognition
IVMS	In Vehicle Monitoring System
LP	License Plate
LPR	License Plate Recognition
LPs	License Plates
LSTM	Long Short-Term Memory
LUT	Look-Up Table
ML	Machine Learning
MCT	Modified Census Transform
MACRNN	Multi-scale Attention CRNN
NLP	Natural Language Processing
OCR	Optical Character Recognition
OF	Optical Flow
PSO	Particle Swarm Optimization
RFID	Radio Frequency Identification
RF	Random Forest
RNN	Recurrent Neural Network
RLR	Red Light Running
ROI	Region Of Interest
SMS	Short Message Service
SORT	Simple Online and Real-Time Tracking
TM	Taguchi Method

TAA	Target Action Analysis
TM	Template Matching
VANET	Vehicular Ad-Hoc Network
VTT	Violation Target Tracking
YOLO	You Only Look Once

List of Symbols

μ	Average value
b	Bias
cX,cY	Bounding box's center
C	Convolution
d	Distance
D	Down sampling
x_j	Feature sequence
mf_4	Final fused feature output to the recurrent layers
f	Function that determines how well feature sequences correlate
G	Gaussians
h	Height
s_{i-1}	Hidden state
x_l	Input feature of residual block *l*
A_i	Intersection area
Σ^{-1}	Inverse of the covariance matrix
α	Learning rate
y_α^t	Likelihood of correctly predicting character
d_m	Mahalanobis distance
T_x	Number of feature sequences
F	Residual function
e_{ij}	Score of the second layer of the neural network
w_l	Set of weights
s	Speed
σ	Standard deviation of the Gaussian distributions
T	Threshold
t	Time
T_d	Time difference
A_u	Union area
w_{21}	Weights of the second layer
w	Width

Chapter One: Introduction

1.1 Introduction

Traffic violations have grown to be a major issue in the modern world. Due to the growing number of vehicles and other vehicles, there is a considerable flow density of traffic [1]. The problem is becoming more severe day by day. The number of accidents in cities is rising. There is obvious need for a control over traffic because drivers are breaking the law [2]. Traffic accidents can cause massive human and material losses [3]. Certain high-risk road segments, like those near schools and major thoroughfares, have traffic infraction monitoring systems installed in order to preserve traffic order and lower the frequency of traffic accidents. Traffic infractions including speeding, unauthorized lane changes, and infractions of traffic signs within the monitored area can all be recorded and handled by a traffic violation monitoring system. It mainly makes use of networking and computer image processing technologies, and an automatic detecting device is used to gather information about unlawful vehicles [4][5]. The impact of traffic infraction monitoring on driving habits and vehicle speed has been investigated by several academics. The likelihood of traffic accidents can be successfully decreased with traffic violation monitoring. Road traffic safety benefits from traffic violation monitoring. Traffic violation monitoring lowers the number of collisions. Automating the procedure is the answer to this issue by using machine learning [6].

1.2 Traffic Violation Detection System

A traffic violation detection system can be equipped with multiple technologies. There are several techniques widely used to detect violations including sensor-based technology backed by Internet of Things (IoT), computer vision, and wireless networks like Radio Frequency Identification (RFID) and Vehicular Ad-Hoc Network (VANET). The optimal performance in its application may come from combining other existing technology classifications [7]. The tools that the system used to detect traffic violations, such as cameras, sensors, In Vehicle Monitoring System (IVMS) Servers.. etc, as shown in Fig. 1.1.

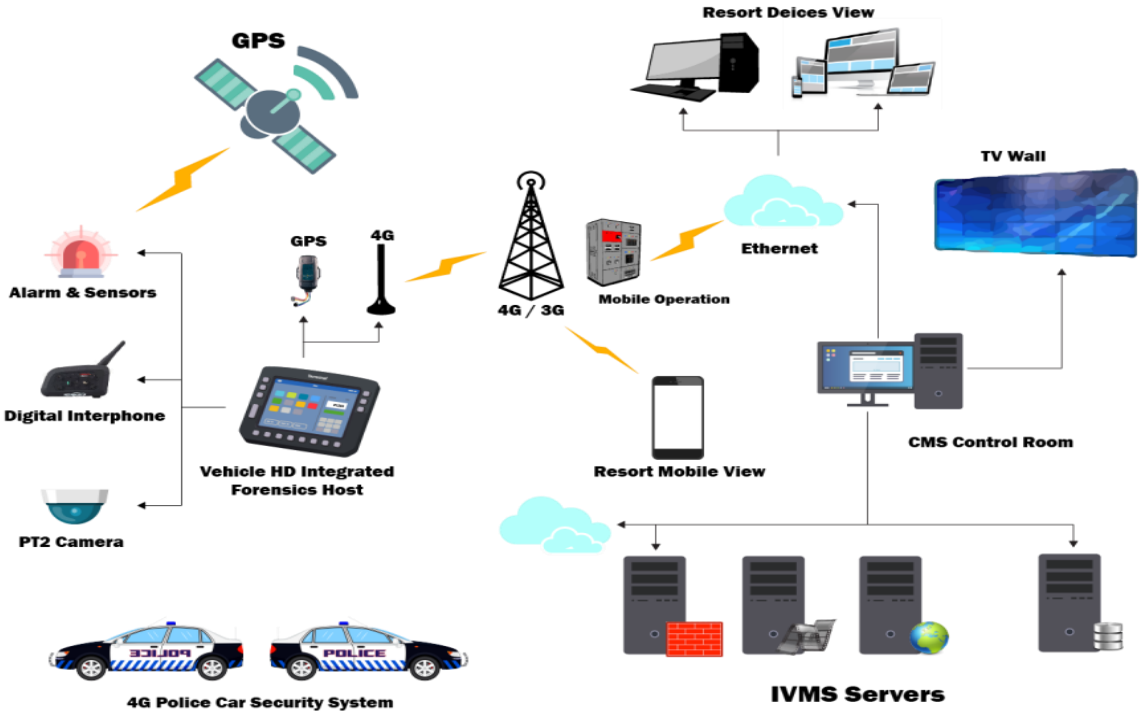


Fig. 1.1 Infrastructure of the vehicle violation detection system [7]

Traffic management systems are deployed to monitor vehicles that violate traffic laws. These procedures need to be automated and made more

efficient. Thus, a simple and efficient vehicle identifying system is required as shown in Fig. 1.2. The crucial query is: "How can a specific vehicle be recognized?". The license plate (LP) is the obvious way to answer this issue because each vehicle has a unique number that makes it easy to identify from other vehicles. Every country assigns its license number to a vehicle displayed on the LP. This number helps to identify one vehicle from another, especially when they are both of the same type and model. An automated system can be applied in place to recognize a vehicle's LP and get the characters and digits from the area where the LP is located. Additional information about the vehicle and its owner can be obtained using the LP number and used for processing. Such an automated system ought to be lightweight, portable, and capable of processing data quickly enough. In recent years, several algorithms for detecting LPs have been created. Every algorithm has benefits and drawbacks of its own [8].

Fig. 1.2 Automatic LP recognition system [8]

1.3 License Plate (LP)

Currently, identifying a vehicle is the primary functional component. A vehicle's LP is the major means of instantaneous identification. Hence, its detection and recognition are regarded as paramount evasion. Even though the LP detection has been a significant topic in numerous research projects over the years, the emergence of new technologies like autonomous driving, the IoT, big data, and deep learning has presented us with several new solutions and obstacles. Therefore, a problem-solving approach that does not require prior limitations must be envisioned. Most existing techniques for identifying LP in photos and movies are only effective in specific, limited settings. One such constraint is the Region of Interest (RoI), which is based on the assumption that the LP is localized in a particular area of the test image frame. Other commonly accepted limitations are camera settings and lighting circumstances. In practical applications, static cameras are often employed in lighting conditions that provide good visibility. The area of the problem is further limited by the aspect ratio of the LP's dimensions, orientation, and color patterns. So, prior understanding of LPs characteristics which existed in different types, as shown in Fig. 1.3, enables of creation a straightforward solutions that take advantage of the real-time performance [9].

Fig. 1.3 Samples of Iraqi LPs

1.4 Problem Statement

The problem of traffic violations encompasses a range of behaviors that deviate from established traffic laws and regulations, posing risks to road safety and the efficient flow of traffic. The definition of this problem involves several key components:

1. **Behavioral Deviation:** traffic violations involve actions taken by road users that contravene established traffic laws and regulations. These behaviors include speeding, running red lights or stop signs, illegal overtaking, and going in the opposite direction.
2. **Safety Risk**: traffic violations increase the likelihood of accidents, injuries, and fatalities on roads and highways. Violations such as speeding and reckless driving can significantly impact the safety of all road users, including pedestrians, cyclists, and other motorists.
3. **Social and Economic Impact**: traffic violations have broader social and economic implications, including costs associated

with healthcare, property damage, and loss of productivity due to traffic congestion and accidents. Moreover, violations can erode public trust in transportation systems and institutions responsible for road safety.

4. **Technological Solutions**: technology advancements, such as automated surveillance systems, vehicle-mounted cameras, and data analytics offer opportunities to enhance traffic enforcement and reduce violations. These technologies can improve the accuracy and efficiency of violation detection while reducing the burden on law enforcement personnel.

1.5 Thesis Aims and Objectives

In accordance with the challenges faced by the latest technologies used to detect traffic violations and recognize LPs, the proposed system aims to automatically detect violations of traffic rules. The system relies on a convolutional neural network algorithm and You Only Look Once Version 4 (YOLOv4) using the camera to achieve the goal of finding the fastest and most accurate algorithm in detecting and recognizing LPs with the following objectives:

- Automatic detection: the device must automatically detect violations without the need for manual intervention, which reduces the time and effort expended in traditional methods.
- High accuracy: the system for detecting violated vehicles must have a high degree of accuracy to correctly identify vehicles and take legal action against them.

- Real-time processing: the violated vehicle plate must be identified at the time of the actual violation for it to be recorded in the message sent to the Traffic Directorate and the vehicle owner.
- The proposed system does not use any sensors to detect traffic violations; instead, a camera connected to a laptop is used to photograph the movement of vehicles, detect violated vehicles and identify their LPs using a Python program.

1.6 Challenges

Automatic traffic violation detection systems are increasingly used to enhance road safety and enforce traffic laws. However, the system faces several challenges:

- **Environmental Conditions:** weather conditions such as rain, fog, or snow, and varying lighting conditions can affect the camera's performance, leading to difficulties in accurately capturing vehicle behavior.
- **Camera Limitations:** high-quality cameras are required for precise detection, but they can be expensive and may have limitations in resolution and range. Additionally, camera placement and angle can impact their effectiveness.
- **Handling of Edge Cases**: situations such as emergency vehicles or unique traffic patterns may not always fit neatly into the system’s algorithms, leading to potential issues with detection and enforcement.

- **Cost:** implementing and maintaining advanced detection systems can be expensive. Budget constraints can limit the extent and effectiveness of deployment.

Addressing these challenges requires ongoing technological advancements, thoughtful policy-making, and a commitment to balancing enforcement with fairness and privacy considerations

1.7 Thesis Outline

The remainder of the thesis is organized as follows:

• Chapter Two covers the main concepts of the YOLO, background subtraction, several Convolutional Neural Network (CNN) models and also includes a literature review of the previously used detection rules, traffic violation and License Plate Recognition (LPR) approaches.

• Chapter Three presents the methodology used for the detection of traffic violation and LPR and includes a detailed explanation of the techniques, algorithms, and processes employed to perform LPR accurately in real time mode.

• Chapter four discusses the results of the used methods and compares them with the techniques described in other works.

• Chapter Five presents the conclusion and recomodations for further future work

Chapter Two: Literature Review and Backgrounds

2.1 Introduction

This chapter collects relevant research from the current literature, which is divided into two parts: Traffic Violation Detection and LP detection and Recognition, and compares these approaches. Furthermore, it comprehensively introduces the fundamental aspects of violated vehicle detection and LP identification and recognition.

2.2 Literature Review

2.2.1 Traffic Violations Detection

Billones et al., 2016 [10]**,** proposed an intelligent system architecture was presented for detecting traffic violations based on computer vision. The algorithms included vehicle detection and tracking using Binary Large Object (BLOB) analysis, Kalman filter, and plate character recognition using Optical Character Recognition (OCR). Traffic violations addressed were number coding, overspeeding, and diverging. The study was in the initial phase, and experimental results indicated 86.11% accuracy for optical character recognition and 88.45% accuracy for speed measurement.

Yim et al., 2020 [11]**,** proposed and evaluated an automatic plate recognition and speed detection system for potential deployment on urban roads. A total of 30 runs were conducted to test the system, with results indicating a 90% success rate in measuring speed, a 56.67% success rate in detecting LPs, and a 36.67% success rate in correctly recognizing LPs. Despite encountering hardware glitches with the sensor during some speed

detection trials, the system showed promise once lessons from unsuccessful trials were incorporated, such as improving character recognition.

Grents, Varkentin, and Goryaev, 2020 [12]**,** conducted a study using a Simple Online and Real-Time Tracking (SORT) tracker along with a two-stage Faster-Region-based Convolutional Neural Network (Faster R-CNN) detector. Their goal was to track, categorize, and determine vehicle speeds. The results of the tests showed that the system could accurately identify, count, and estimate vehicle speeds with an error rate of less than 22%. The system achieved an accuracy rate of 78% in classifying vehicles and estimating their speeds.

M. M. Bachtiar et al. 2020 [13], suggested a system works for vehicle detection and traffic violation detection the mean shift algorithm was used for vehicle tracking and BLOB technology to detect vehicles violating traffic signals, and the system achieved a detection accuracy of 71%.

Wankhede & Bajaj 2021 [14]**,** addressed traffic problems in India, emphasizing diverse traffic causing congestion and essential accidents. Social factors like jumping traffic signals and mobile phone use were significant contributors to accidents. The paper focused on developing a behavioral detection model using a matching approach and Particle Swarm Optimization (PSO). Testing on Nagpur city traffic videos showed increased accuracy in detecting violations. The model specifically targeted red-light violations, and mobile phone usage, crucial factors in road accidents, showcasing notable red-light violations were detected using the matching techniques and PSO algorithms; experimental results showed accurate accuracy of 89.6% and 91.1% were attained, respectively.

Anand, Kilari and Kumar, in 2021 [15]**,** proposed a system that highlighted the surge in new vehicles, contributing to congested roads and

increased traffic violations. To address this, a computer vision-based traffic violation detection system was proposed and implemented using You Only Look Once version 3 (YOLOv3). The system focused on detecting violations like signal jumps, vehicle speed, and count, achieving an accuracy of 97.67% for vehicle count and 89.24% for speed detection. The authors suggested YOLOv3's suitability for traffic violation detection, particularly in handling multiple violations from a single input source. The system operated efficiently, with speed dependent on traffic density, demonstrating lower detection times in high-density traffic flows.

Peng et al., 2022 [16], aimed to create an automatic traffic violation detection system using videos reported by the public, especially those from dashcams. They focused on two violations: running a red light and turning on a red light. The system included Violation Target Tracking (VTT) and Target Action Analysis (TAA), utilizing depth and gradient angle changes (GAC). The experimental results showed an average accuracy of 76.1% and a conditional accuracy of 81.9%. The developed system aimed to alleviate law enforcement's resource demands in processing the growing number of reported traffic violations.

M. Zahid et al., 2022 [17]**,** proposed a system to detect Red Light Running (RLR) violations using spatial analysis and Machine Learning (ML) methods. Data imbalance was addressed using random over-sampling. Random Forest (RF) and Gradient-Boosted Decision Tree (GBDT) algorithms classified and predicted RLR, with GBDT achieving 96% accuracy and outperforming RF.

T. Singh et al., 2023 [18], introduced an improved traffic light violation detection method, utilizing a modified YOLO algorithm and Hough space analysis (HSA). The algorithm, implemented using Apache Kafka and

Apache Spark, operates autonomously, adapting to environmental changes. It achieved 88.24% accuracy in detecting traffic light violations, showing scalability and effectiveness in real-world scenarios. The method combines lightweight YOLO for object detection and Hough space analysis to identify violation regions during red lights, demonstrating suitability for real-time traffic violation detection.

Table 2.1 Summary of the related work regarding the violation detection

Ref. No.	Year	Method	Violation Type	Accuracy rate (%)
[10]	2016	BLOB analysis and Kalman filter	Exceeding the speed limit	86.11
[11]	2020	Sensors	Exceeding the speed limit	90
[12]	2020	SORT	Exceeding the speed limit	78
[13]	2020	BLOB	Traffic signal	71
[14]	2021	PSO	Traffic signal	91.1
[15]	2021	YOLOv3	Exceeding the speed limit	89.24
[16]	2022	Depth and GAC	Traffic signal	81.9
[17]	2022	RF+GBDT	Traffic signal	96
[18]	2023	YOLO and HSA	Traffic signal	88.24

2.2.2 LPs Detection and Recognition

Xie et al., 2018 [19], proposed a technique for detecting LP and recognizing characters. It is based on a combination of a feature extraction model and a Backpropagation Neural Network (BPNN) and can adjust to complex backdrops and weak illumination. The approach comprised candidate region verification, contrast enhancement preprocessing, and

precise LP localization using the integral projection method. Through BPNN, the feature extraction model uses three feature sets trained to recognize characters accurately. The experimental results demonstrated a 97.7% recognition accuracy.

Altameemi, Abbas and Alabaichi, 2019 [20], proposed a system that confirmed that the need for recording vehicle movements in Iraq was to address security concerns and support various sectors like statistics, economics, and tourism. The algorithm was developed and tested using the Matlab package to leverage Graphics Processing Unit (GPU) capabilities for handling extensive data and matrices. An algorithm was implemented to detect, extract, localize, and recognize Iraqi vehicle registration plates, utilizing object-feature and template-matching (TM) techniques. The method achieved 96% accuracy in recognizing numbers.

Hendry and R. C. Chen 2019 [21]**,** conducted Automatic License Plate Recognition (ALPR) using the YOLO-darknet deep learning framework. The algorithm utilized sliding window detection to recognize vehicle LPs, specifically from Taiwan. The system achieved 98.22% accuracy in LP detection and 78% accuracy in license plate recognition. The study underscores the effectiveness of YOLO-based approaches in ALPR tasks, particularly in accurately detecting and recognizing LP under diverse environmental conditions.

Silva and Jung, 2020 [22], emphasized The ALPR system and its importance in various areas like Intelligent Transportation and Surveillance systems. They showcased a sophisticated approach using a hierarchical Convolutional Neural Network (CNN) for end-to-end ALPR. They developed two CNN networks based on YOLO: a Large-scale Place Description Network, which could identify frontal/rear views of vehicles along with LPs

in a cascaded manner, License Plate Segmentation, and a Character Recognition Network, which focused on detecting characters within cropped LP regions. Testing their method on datasets containing European and Brazilian license plates yielded an impressive 89.15% accuracy in identifying vehicle LPs using CNN and OCR algorithms.

Anson and Mathew, (2020) [23], highlighted the significance of ALPR, a technique crucial for tracking, identifying, and monitoring moving vehicles by automatically recognizing LPs through image processing. They employed a CNN to scan the entire image for character recognition. Further, they used a plate/non-plate CNN classifier to filter out non-plate areas from the identified plate region. By combining CNN with Long Short-Term Memory (LSTM) approaches, they claimed to achieve an accuracy rate of 85.0%.

D. A. Abd Alhamza and A. D. Alaythawy , 2020 [24], developed The Iraqi License Plate Recognition (ILPR) system to recognize numbers using a combination of pre-processing, segmentation, and K-Nearest Neighbors (KNN). This system effectively spots and identifies Arabic numbers on LPs, boasting an 90% accuracy rate. They built it using Python3.5 and the OpenCV library. The system works through three key stages: pre-processing, where the image is prepared; LP detection and segmentation, which isolates the plate from the rest of the image; and finally, number recognition, ensuring precise identification.

X. Zhang et al. 2021 [25], explored machines and deep learning algorithms, such as CNN, LSTM, and KNN, to tackle the detection of Chinese LPs. When comparing different models using the Chinese City Parking Dataset (CCPD) dataset, the CRNN model stood out 95% accuracy rate. This

underscores CRNN's effectiveness in pinpointing Chinese LPs accurately, especially within the Chinese City Parking Dataset.

Khan et al., 2021 [26], proposed a method for recognizing multiple LPs in high-resolution images. Their technique involved a two-step process for plate detection. Initially, all vehicles in an image were identified using the Faster-Region-based Convolutional Neural Network (Faster R-CNN), providing accurate data to pinpoint LPs. Then, plates were localized in the HIS (hue (H), saturation (S), intensity (I)) color space using geometric attributes and morphological techniques to filter out non-plate regions. They employed a Look-Up Table (LUT) classifier with adaptive boosting and the Modified Census Transform (MCT) as a feature extractor to recognize characters. The results were impressive, boasting an overall detection rate of 96.72% and a recognition rate of 98.02% for numerous LPs, as per their experiments.

I. Mihoub (2023) [27], proposed a deep learning-based LP detection and recognition system that utilizes YOLO and CNN for detecting and recognizing LPs, achieving 87% and 93% accuracy rates, respectively.

Table 2.2 Summary of the related work regarding the recognition of LPs

Ref. No.	Year	Method	Accuracy rate (%)
[19]	2018	BPNN	97.7
[20]	2019	TM	96
[11]	2020	OCR	56.67
[21]	2019	OCR	78
[22]	2020	CNN+OCR	89.15
[23]	2020	LSTM+CNN	85

[24]	2020	KNN	90
[25]	2021	CRNN	95
[26]	2021	Faster R-CNN	98.02
[27]	2023	CNN+YOLO	93

2.3 Vehicle Detection

For the effective administration of transportation networks, road safety, and public safety on crowded streets and highways, private businesses, governments, and public organizations use traffic surveillance. One typical example of a surveillance system is a stationary camera monitoring a scene. Finding invasive items is a crucial first step in scene analysis [28]. Once moving foreground objects are successfully separated from the background, object classification, vehicle identification, tracking, and activity analysis are all made easier. Moving object segmentation offers the ease of object-based representation and manipulation of video content by removing relevant information about the moving vehicle from video sequences. For many computer vision applications, including pattern recognition, video compression, video retrieval, and video surveillance, it is a crucial step. Optical Flow (OF), background subtraction (BS) , and frame difference methods are examples of conventional techniques for segmenting moving objects [29]. The methods used in motion detection, frame differencing, OF and BS methods will be explained in some detail.

2.3.1 Frame differencing

In order to identify regions that correspond to moving objects, such as people and cars, frame differencing is the process of pixel-by-pixel comparison between two or three successive frames in a picture stream. Change is determined by the threshold function, which is based on how quickly an object moves. It is hard to keep the segmentation quality high if the object's speed changes dramatically. The inter-frame differencing method compares two consecutive frames to identify segments of moving objects. However, because it can only distinguish variations in the background, it can only recognize portions of a vehicle that was partially hidden by the background in the preceding frame. This system is not able to handle realistic traffic situations where vehicles may stop for extended periods of time, even with some improving techniques [30].

2.3.2 Optical Flow (OF)

OF employs the flow vectors of the moving objects over time to identify moving regions in a picture. Applications involving motion-based segmentation and tracking make use of it. Each pixel region's translation is defined by a dense field of displacement vectors. When there is a motion in the camera, OF works best. The popular flow calculation methods are computationally complex [31].

2.3.3 Background Subtraction Method (Gaussian Mixture Model (GMM))

In the world of computer vision, understanding what's moving and what's not is vital, especially in fields like surveillance, tracking objects, and recognizing activities. The main aim here is to accurately pick out the things that are on the move or catch our interest from the background that stays the

same in a video. Having strong and effective techniques for background modeling and foreground separation is crucial for the success of many computer vision tasks[32].

Most Background Subtraction methods follow a similar technique. It includes two significant steps: background initialization with maintenance and foreground detection. Background initialization constructs an initial background model according to a specified number of frames. In foreground detection, a comparison is made between the current frame and the background model for each frame, leading to calculating the scene's foreground. Typically, the results of the foreground detection are fed back into the background model for updating [33].

A more robust method is to model the background with Gaussian Mixture Model (GMM). In this algorithm, the distribution of the color of each pixel is modeled as a sum of weighted Gaussian distributions defined in a given color space. Namely, the intensity is modeled as [34]:

$$P\big(I(x,y,t)\big) = \sum_{i=1}^{K} w(i,x,y,t).\, G\big(I(x,y,t), \mu(i,x,y,t), \sigma(x,i,y,t)\big) \qquad \textbf{(2.1)}$$

At any time t , what is known about a particular pixel is its history. For each Gaussian function, two parameters $(\mu\ , \sigma)$ need to be stored. Parameter μ is estimated as the average value of previous pixels. The parameter σ is the standard deviation and is estimated through previous pixel samples as well. The history is modeled by a mixture of K Gaussians G. Thus, the probability of occurrence of an intensity at a given pixel (x,y) is represented as in Eq. (**2.1**). The weight of the ith Gaussian model, G, is $w(i,x,y,t)$. The GMM may simulate appearance as a combination of many items. A backdrop pixel is best portrayed as a combination of many (typically tiny) objects if the background

features moving textures, such as sea waves, wind-shaking tree branches, or even merely fluctuations in lighting throughout the day. Initialization of the means, standard deviations, and mixing weights the parameters in the Gaussian functions is required. They may be initialized at random. However, the standard deviation ought to be sufficiently high, and the emerging Gaussian's weight ought to be sufficiently low. A higher standard deviation allows for the inclusion of more data in a single model, leading to a more accurate model. A different method of initialization is to use the kmeans findings from training samples. It can be calculated from the same cluster, and the corresponding weight is the ratio of samples from this cluster. Can set the value of as the average value for one cluster obtained by kmeans. Pixels are then classified as "in motion" if they deviate from any of the calculated Gaussian distributions by more significant than (2 or 3) standard deviations [34].

$$|I(x,y,t) - \mu(x,y,t)| > k\sigma(i,x,y,t-1) \quad \textbf{(2.2)}$$

If a new pixel value $I(x,y,t)$, can be matched to one of the existing Gaussians (within $K\sigma$)),then the matched Gaussian $\mu(i,x,y,t)$ and $\sigma\ (i,x,y,t)$ are updated as follows [34]:

$$\mu(i,x,y,t) = (1-\rho)\mu(i,x,y,t-1) + \rho I(x,y,t) \quad \textbf{(2.3)}$$

$$\sigma^2(i,x,y,t) = (1-\rho)\sigma^2(i,x,y,t-1) + \rho\big(I(x,y,t) - \mu(i,x,y,t-1)\big)^2 \quad \textbf{(2.4)}$$

where α is a learning rate and $\rho = \alpha$ G($I(x,y,t), (i,x,y,t), \sigma\ (i,x,y,t)$) The previous weights of every Gaussan are modified as follows [34]:

$$w(i,x,y,t) = (1-\alpha)w(i,x,y,t-1) + \alpha(M(i,t)) \quad \textbf{(2.5)}$$

where $M(i,t)$= 0 for all other values and $M(i,t)$ = 1 for the matched Gaussian. Objects are permitted to blend into the backdrop under this model. In the event that $I(x,y,t)$ does not correspond to any of the K known Gaussians, a new distribution is substituted for the least likely one. The word "least probable" has a w/σ meaning. The new distribution has low weight, high σ, and the formula $\mu(i,x,y,t)$= $I(x,y,t)$. Following the updating of each Gaussian, the K weights $w(i,x,y,t)$ are normalized till their total equals 1. The Gaussians with the highest supporting weight and lowest variance should be matched by the backdrop. A larger weight indicates a broader support zone, and a lower variance indicates a higher probability. The distribution with the lowest variance, greatest probability, and largest area of support is thought to be a component of the background. Subsequently, the least number b of distributions whose weights add up to a sizeable enough fraction of space T is chosen as the background model, as in Eq. (2.6) [34].

$$B = argmin_b\left(\sum_{i=1}^{b}\frac{w_i}{\sigma_i} > T\right) \quad \textbf{(2.6)}$$

where T is a threshold. The background subtraction result is shown in Fig. 2.1 for the same testing frame.

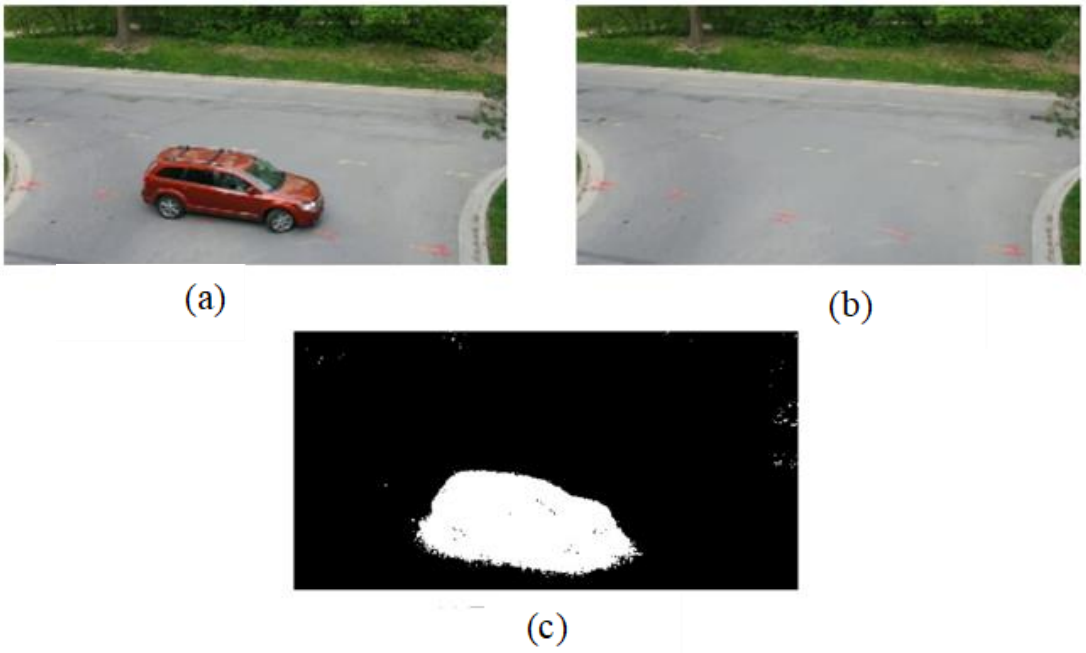


Fig. 2.1 BS using GMM (a) Input video (b) Background model (c) Foreground mask [34]

In the foreground mask, noise is still there. However, the GMM approach does the best at obtaining the whole foreground object out of all of these techniques. However, it performs significantly worse than the mean Gaussian average technique in terms of providing precise data at the object border. Even if a different threshold is selected for each pixel using the Gaussian mixture model and Gaussian average, and these pixel-wise thresholds are modified over time, it does not consider spatial information. That is, each pixel makes a decision independently of other pixels, even though decisions are correlated. That is why the foreground mask boundary is not accurate even in GMM [32].

2.4 Morphological Operations

Grayscale images can be subjected to morphological processes since their absolute pixel values are deemed insignificant or of small interest, and their light transfer functions are unknown. Morphological approaches analyze a picture that contains a structural element, which is a small form. A binary picture, or small matrix of pixels with a value of either one or zero for each pixel, serves as the structural element [35]. Out of many morphological operations, some are described as follows.

2.4.1 Dilation

The dilation process causes an object's size to increase. The type and form of the supporting element determines how much it expands. The definition of an image *A*'s (set) dilatation caused by structuring element B is defined as [36]:

$$A \oplus B = \left\{ Z \middle| (\widehat{B})_Z \cap A \neq \emptyset \right\} \quad \textbf{(2.7)}$$

The dilation of A by B is the set of all displacements Z such that $\hat{B}$ and A have at least one element in common if set B is moved and reflected about its origin by Z . As previously stated, dilation increases the boundary elements' pixel count. The process of dilation causes the number of pixels with value zero (background) to decrease and the number of pixels with value one (foreground) to increase. Additionally, a continuous object's holes or missing pixels are essentially filled in via the dilation technique. The dilation operation alters the intensity at the object's boundary by adding pixels, and as a result, a blurring effect is visible. Thus, it can be compared to spatial low-pass filters

that are smoothed while applying linear image filtering image, as shown in Fig. 2.2 [36].

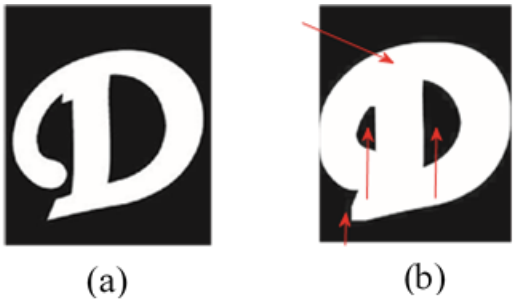


Fig. 2.2 Morphological dilation (a) Original image (b) Dilated image [37]

2.4.2 Erosion

In relation to the operation impact, the erosion operation is the dilation operation's complement. That example, the process of erosion results in an object losing size. The definition of an image A being eroded by a structural element B is [36]:

$$A \ominus B = \{Z|(B)_Z \subseteq A\} \quad \textbf{(2.8)}$$

The set of all places Z such that the structuring element B is translated by Z is a subset of the picture is the erosion of image A by structuring element B. The object's boundary pixels are lost as a result of this action. The number of pixels with value zero (background) increases while the number of pixels with value one (foreground) decreases as a result of erosion. Structures that are smaller than the structuring element are eliminated by the erosion process. Therefore, the noisy "connection" between two things can be eliminated using it. When undesirable pixels are "erased," the item in a picture becomes sharper overall. The sharpening of a high pass filter, which is utilized in linear image filtering, is comparable to the erosion procedure, as shown in Fig. 2.3 [36].

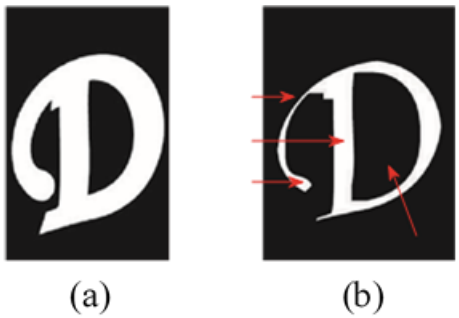


Fig. 2.3 Morphological erosion (a) Original image (b) Erosion image [37]

2.4.3 Closing

Additionally, erosion and dilation work together to close a picture. In terms of the sequence in which erosion and dilation occur, it is different from the opening procedure. The definition of a structural element B 's closing of the picture A is [37]:

$$A \cdot B = (A \oplus B) \ominus B \qquad \textbf{(2.9)}$$

The mathematical statement (2.9) illustrates the relationship between erosion and dilation with closure. It demonstrates that a closure operation is when a structuring element B dilates an image A and then erodes the resulting image using the same structuring element. The points in the structuring element B that, when B is "rolled" over A around outside of its border, reach the extreme points of A's boundary constitute the closed image's boundary. Even if the closure procedure smoothes some curves, it generally combines thin gaps and narrow breaks. Consequently, it eliminates tiny gaps and fills in the edges of the objects, as shown in Fig. 2.4 [37].

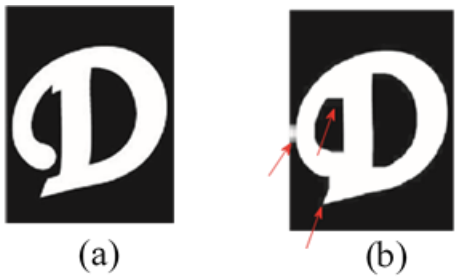


Fig. 2.4 Morphological close (a) Original image (b) Close image [37]

2.4.4 Opening

The opening of an image is a combinational operation of erosion and dilation. The opening of image *A* by structuring element *B* is defined as:

$$A \circ B = (A \ominus B) \oplus B \qquad \textbf{(2.10)}$$

The above definition gives the relationship between opening and erosion & dilation. It states that the opening operation is nothing but the erosion of an image by a structuring element and the resultant is dilated with the same structuring element. The boundary of the opened image is the points in the structuring element *B* that reaches the extreme points of the boundary of *A* as *B* is 'rolled' around inside of this boundary. The union set operation is also used in literatures to find the points of the opened image. The opening operation smoothest the outline of an object clears the narrow bridges and also eliminates minor extensions present in the object, as shown in Fig. 2.5 [37].

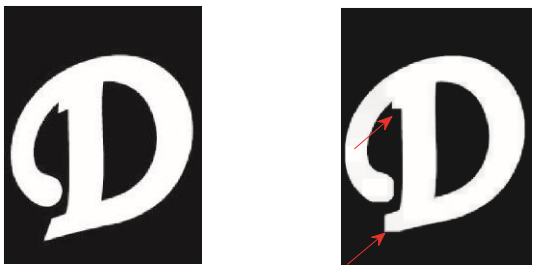


Fig. 2.5 Morphological open (a) Original image (b) open image [37]

2.5 You Only Look Once (YOLO)

YOLO is an object detection algorithm used in computer vision applications. It has gone through version different iterations, each one an improvement over the previous version.

YOLO has undergone several changes and improvements to incorporate the latest developments in computer vision research. Some concepts were dropped since the algorithm could not perform and be accurate enough. These adjustments and improvements have made YOLO one of the most popular and reliable object detection systems currently available [38].

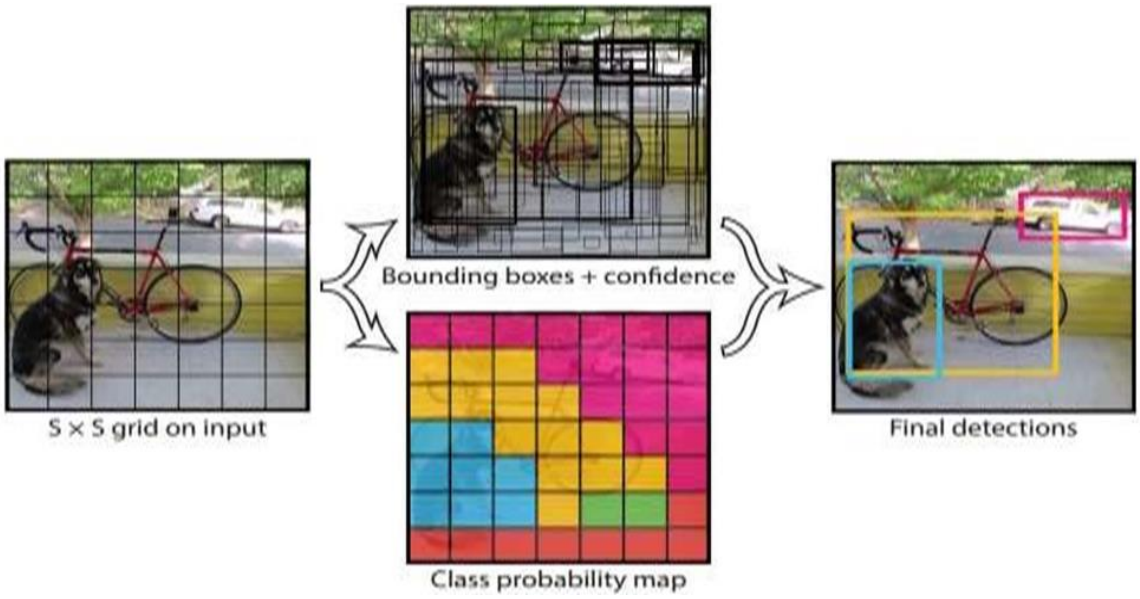


Fig. 2.6 YOLO'S Summary [39]

This algorithm's operation is predicated on the division of the image into individual grid cells Side-by-Side (S×S) as shown in Fig. 2.6. One needs to know what to expect before delving into the specifics of the YOLOs. The aim is to forecast the object's class and bounding box, which indicate the

object's location. Several descriptors can be used to describe each of the bounding boxes.

- Width (w)
- Height (h)
- The bounding box's center (cX and cY).
- The letter c stands for an object class, such as a car or traffic lights.

Furthermore, an extra predicted value, pc, represents the probability that an object is inside the bounding box as shown in Eq. (2.11). The object detection method is a single regression problem that goes straight from the image pixels to the bounding box coordinates and class possibilities, as seen in Fig. 2.7.

$$LP = (pc, cX, cY, w, h, c) \quad \textbf{(2.11)}$$

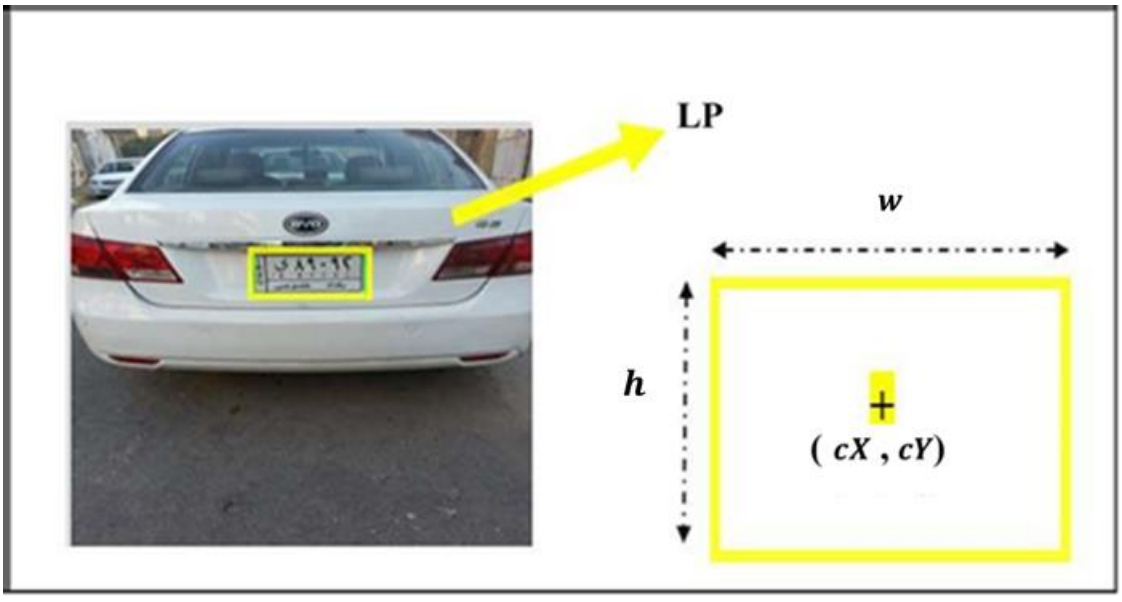


Fig. 2.7 Bounding box [40]

2.5.1 Locating the LP Within the Image

The current study employs a single CNN to localize an LP through the bounding box automatically. The bounding box center coordinates (cX, cY) are predicted using the YOLOv4 algorithm along with its width (w) and height (h) . Equations (2.12) and (2.13) are used to derive the bounding box corners at the left side [41].

$$X_0 = cX - (w / 2) \quad \textbf{(2.12)}$$

$$Y_0 = cY - (h / 2) \quad \textbf{(2.13)}$$

The top right coordinates (X_1, Y_1) of the bounding box are estimated using Eq. (2.14) and (2.15).

$$X_1 = X_0 + w \quad \textbf{(2.14)}$$

$$Y_1 = Y_0 + h \quad \textbf{(2.15)}$$

2.5.2 Intersection Over Union (IoU)

The object detector's accuracy on a particular data-set is measured using an evaluation metric called IoU. The following actions are necessary in order to apply the IoU for assessing a (random) object detector [42]:

1. The ground-truth bounding box, which is a manually labeled box from the testing set that indicates where an object is located in an image.
2. The suggested model's anticipated bounding boxes IoU can be used when there are five groups of bounding boxes. The IoU can be calculated as follows:

$$IoU = \frac{A_i}{A_u} \tag{2.16}$$

where A_i and A_u are the intersection area and union area, respectively; their definitions are illustrated in Fig. 2.8.

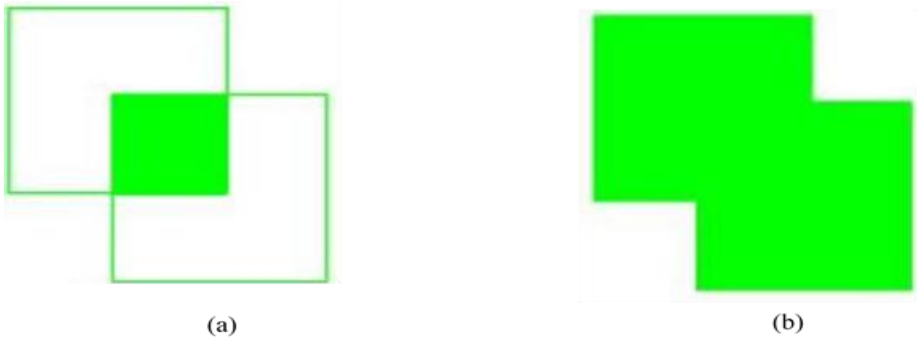


Fig. 2.8 Regions used in calculating IoU (a) Intersection area and (b) Union area [42]

It should be noted that IoU must be any value between 0 and 1 [42]. To predict bounding boxes accurately, the IoU between the expected and actual bounding boxes must be close to 1, as illustrated in the examples shown in Fig. 2.9.

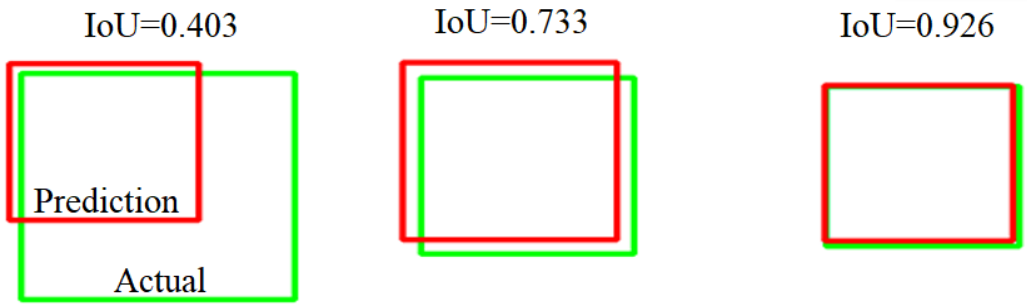


Fig. 2.9 Examples of the IoU values [42]

2.6 The multi-scale attention CRNN (MACRNN)

Let us explore the structure of the multi-scale attention CRNN network. It is a blend of CNN, RNN, and Connectionist Temporal Classification (CTC) orchestrated by an end-to-end text line recognition

algorithm. The identification network can be divided into three main sections, as shown in Fig. 2.10, convolutional layers, recurrent layers, and transcription layers, in sequential order from left to right. The CNN layers, which are made up of feature reuse networks and asymmetric convolution layers, constitute the foundation of the entire network. The feature sequences from various convolution layers of each input image are automatically extracted using CNNs. Next, each frame of the feature sequence that was taken out of the CNNs is predicted by the RNNs. The transcription layer, the final component of the network, is in charge of translating RNN predictions into actual labels [43].

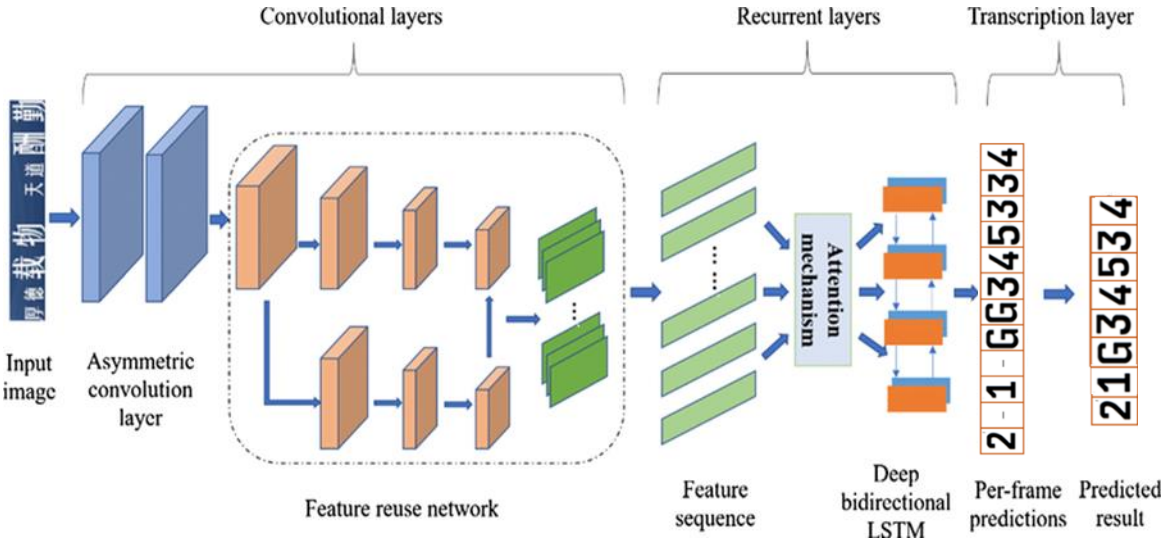


Fig. 2.10 The network structure of MACRNN [43]

The entire network comprises three main parts:

- Through convolutional layers consisting of an asymmetric convolutional layer and a feature reuse network, the feature extraction process is performed.

- Recurrent layers convert Feature sequences into per-frame predictions using an attention mechanism and a bidirectional LSTM.
- The transcription layer generates final predicted labels based on the per-frame predictions of recurrent layers.

2.6.1 Asymmetric convolution

Since text recognition requires identifying a whole text line, input images for text recognition are typically far wider than they are taller than they are for normal image identification. For instance, the average width is 210 pixels when all of the image heights are reduced to 32 pixels. Consequently, the image's breadth is significantly more than its height. But because the width and height of the receptive field of common convolution are the same, it might not be able to extract the text image's features very well. To lower computational costs without sacrificing performance, we therefore apply a horizontal asymmetric convolution operation, which was originally suggested in conception architecture [44]. The network may then adjust itself more to the properties of text line images. The outcomes of the experiment suggest that asymmetric convolution can greatly raise text line identification accuracy. For convolution operation, when 1×3 and 2×3 asymmetric convolution kernels are added to the CNN, as shown in Fig. 2.11. By performing this action, the network improves its capacity to extract information from horizontal text sections by expanding the receptive field of such regions in the image.

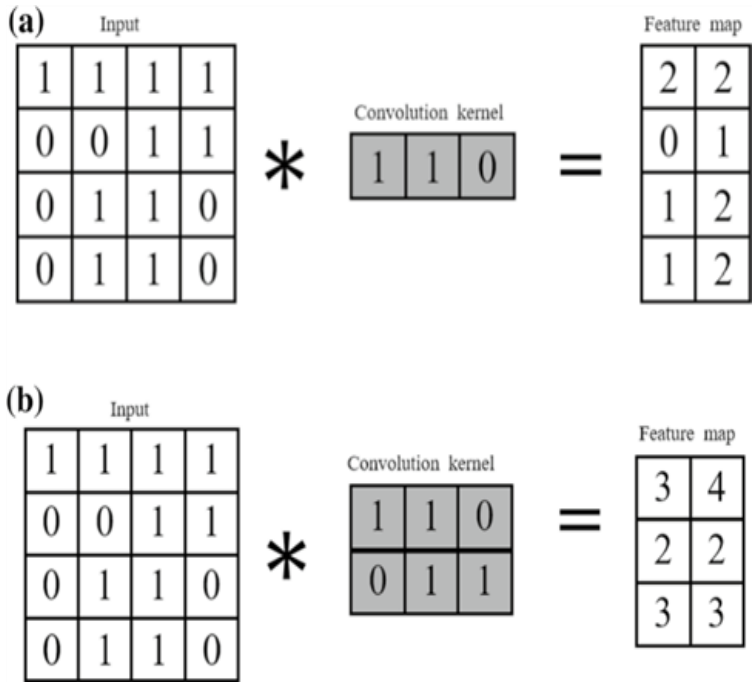


Fig. 2.11 Asymmetric convolution operation (a) 1 × 3 convolution (b) 2 × 3 convolution [43]

2.6.2 Feature reuse network

Although the multiple convolutions and pooling operations in a CNN can extract more abstract, high-dimensional features, image details are more easily lost with an increase of network depth. It is necessary to input a picture and return numerous prediction results in order to recognize a text line in its entirety. In the event that there is a single input and several outputs, the detail feature is especially crucial. When only high-level features are employed, it is simple to lose the feature information of small characters, for instance, if there are varied character sizes in a text line. Therefore, we use a convolutional neural network to carry out a feature reuse operation. The detailed architecture of the feature reuse network is depicted in Fig. 2.12. ResNet50 is the basis for modifications made to the entire feature reuse network .

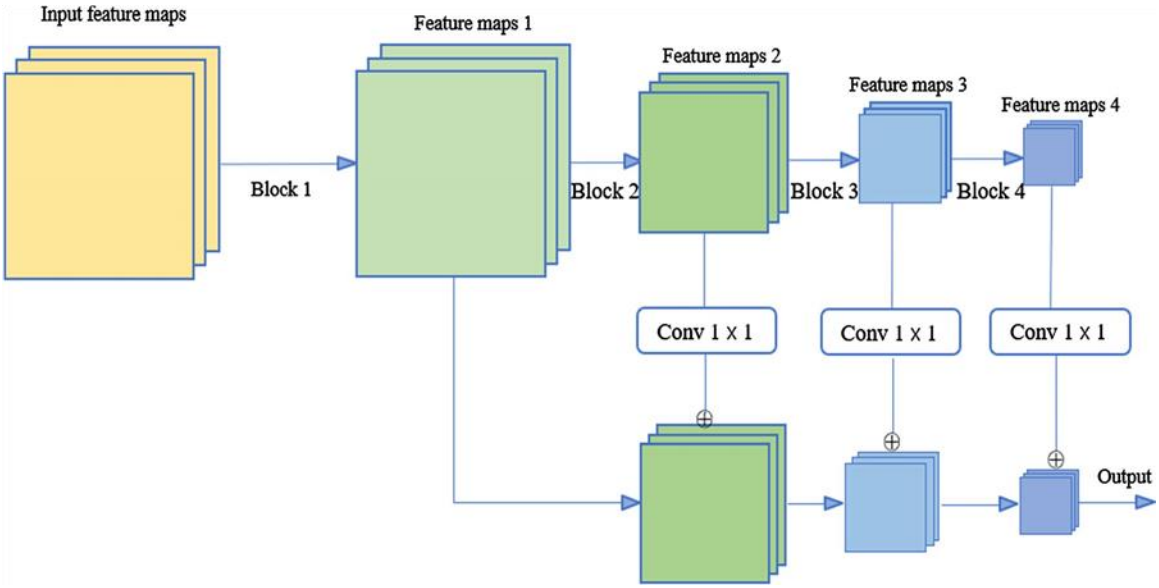


Fig. 2.12 Particular structure of the feature reuse network [43]

The symbols of Block1, Block2, Block3 and Block4, respectively, represent conv2x, conv3x, conv4x and conv5x in ResNet50 [45]. The downsampling is a max pooling operation. Conv 1×1 represents 1×1 convolution operation so as to match dimension when carrying out feature fusion. The symbol $\oplus$ is an element-wise addition operation.

The feature reuse network integrates features from several layers in the network, not simply the last layer's characteristics, as shown in Fig. 2.12. The output features of Block ($i = 2, 3, 4$) are defined as f_i ($i = 2, 3, 4$) while the used features are defined as mf_i ($i = 2, 3, 4$). The feature reuse network's computational procedure can be expressed as [43]:

$$mf_i = D(mf_{i-1}) \oplus C(f_i) \tag{2.17}$$

where $mf_1 = f_1$, the operation of D is down sampling, or max pooling and i = 2, 3, 4; the operation of C is 1×1 convolution, which is in charge of

dimension matching; and the element-wise addition operation is represented by the symbol $\oplus$. mf_4 is the final fused feature output to the recurrent layers that follow. It is evident that the feature reuse network uses a bottom-up cascaded feature reuse module to mix high-level semantic characteristics from the deep network with low-level information from the shallow network. This feature reuse network can significantly enhance text line identification performance and be more resilient to the size and distortion of various characters in the image because the much shallower layers can preserve more specific information of the original image. Fig. 2.13 depicts the block's particular structure, which is a residual connection with the same scales that is defined by Eq. (2.18) [43].

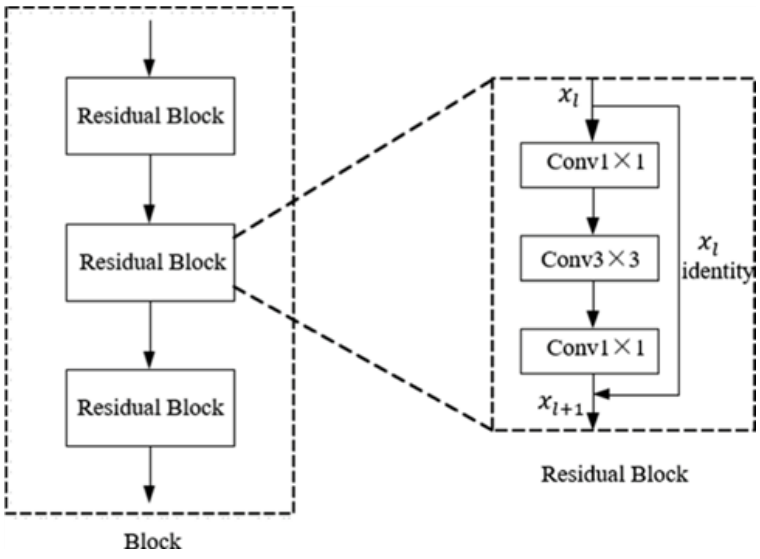


Fig. 2.13 Specific architecture of the residual block [43]

The number of residual blocks of Block1–Block4 is separate from each other After each block, the feature map size is reduced to half and the number of feature maps is doubled [43].

$$y_l = x_l + F(x_l, w_l) \quad \textbf{(2.18)}$$

where x_l is the input feature of residual block l, w_l is a set of weights and bias related to residual block l, F is a residual function. The residual block, as shown in Fig. 2.13, solves the deep neural network's gradient disappearance issue by establishing identity mapping, which directly transfers prior knowledge to the later layer. Furthermore, for network topologies with a given depth, the residual block widens and deepens the network structure, improving the capacity to describe a complex function. By combining low-level and high-level features, the feature reuse network enhances the ability to express features. Consequently, it helps to overcome a variety of difficult circumstances and extract more expressive qualities (e.g., varying sizes of characters, image distortion, blur and illumination) [43].

2.6.3 Recognition Components

Three components make up the entire recognition component: the bidirectional Long Short-Term Memory (LSTM), the CTC layer, and the attention model [46] [47]. In order to produce more accurate predictions, the attention model can fully account for character relationships and contextual information. In the feature sequence $X = x_1, \ldots, x_{T_x}$, the bidirectional LSTM may predict one result y_i for each feature sequence, xi. After that, y_i is changed by CTC layer.

2.6.3.1 Bidirectional LSTM with attention

In text recognition, the RNN can be regarded as the decoder module to decode feature sequences from the CNN and output final prediction results. When studying the encoder–decoder model of neural machine translation (NMT), According to the work introduced in [46], encoding a sentence with a fixed-length vector would result in a significant over-fitting issue,

particularly for lengthy input sequences. This is because, regardless of the length of the input sequence, it is encoded as a fixed-length vector representation; yet, the input fixed-length vector will impose limitations on the decoder throughout the decoding process. The attention mechanism is then put forth as a solution to this issue. In addition, augment a bidirectional LSTM with the attention mechanism. The bidirectional LSTM can therefore receive the feature sequences associated with the current output and pay greater attention to those significant feature sequences in order to improve recognition performance to the attention mechanism, as shown in Fig. 2.14.

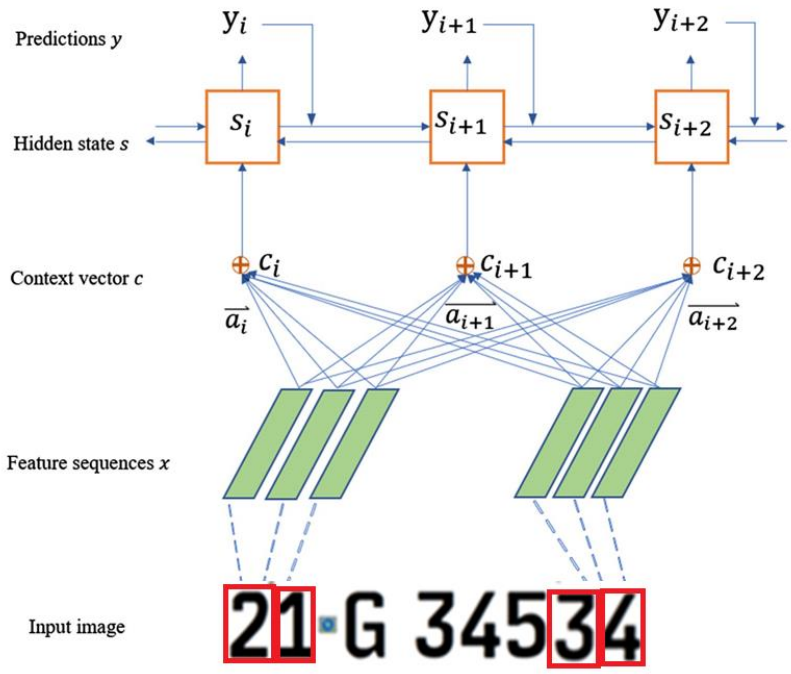


Fig. 2.14 Bidirectional LSTM with attention: structure and contextual summarization [43]

The context vector (c_i) is obtained using Eq. (2.19) [43].

$$c_i = \sum_{j=1}^{T_x} a_{ij} x_j \tag{2.19}$$

where x_j designates as the feature sequence, a_{ij} as the corresponding weights for vector x_j at time i, and T_x as the number of feature sequences. Hence, a_{ij} is denoted by:

$$a_{ij} = f(s_{i-1}, x_j) \tag{2.20}$$

where the feature sequence x_j and the hidden state s_{i-1} of the RNN are used to compute a_{ij}, f is a function that determines how well feature sequences correlate. Since f requires a neural network model, a choice is made to utilize a three-layer neural network with the tanh function serving as the activation function. The method of calculation is as follows [43]:

$$h_{ij} = tanh(w_{11} \times x_j + w_{12} \times s_{i-1} + b) \tag{2.21}$$

$$e_{ij} = h_{ij} \times w_{21} \tag{2.22}$$

where b and w_{21}, respectively, indicate the bias term and the weights of the second layer, e_{ij} indicates the score of the second layer of the neural network, and h_{ij}indicates the output vector of the first layer for the feature sequence x_j and s_{i-1}. After that, e_{ij} is covered with a softmax layer to produce a_{ij}, as in Eq. (**2.23**) [43].

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{K=1}^{T_X} \exp(e_{iK})} \tag{2.23}$$

Following that, the bidirectional LSTM receives the context vectors produced by each attention module and uses them to generate per-frame predictions, which are subsequently sent to the CTC layer.

2.6.3.2 CTC Layer

The Bidirectional LSTM's per-frame predictions are converted to final actual labels using CTC [43]. The goal is to ascertain the greatest likelihood given each prediction. Based on per-frame predictions $Y = y1, \dots, y_{T_x}$, where the conditional probability for label sequence equals l, the probability is calculated as in Eq. (**2.24**).

$$p(l|Y) = \sum_{\pi:B(\pi)=l} p(\pi|Y) \tag{2.24}$$

where B shows a mapping relationship from π to l that removes the repeated labels when mapping π to l and then removes the "blank" labels. Whereas π is created by assigning a label to each time step and concatenating the labels to produce a label sequence.

All conceivable label sequences that B can map to l are represented by π: $B(\pi) = l$. The definition of the probability of π is [43]:

$$p(\pi|Y) = \prod_{t=1}^{T_x} y_{\alpha}^{t} \tag{2.25}$$

where y_{α}^{t} indicates the likelihood of correctly predicting character α in set L at time t (which includes all task labels as well as a "blank" label, such as L = 6736 in Community Team Learning Disability (CTLD) service). Fig. 2.15 shows a clear explanation of the CTC layer.

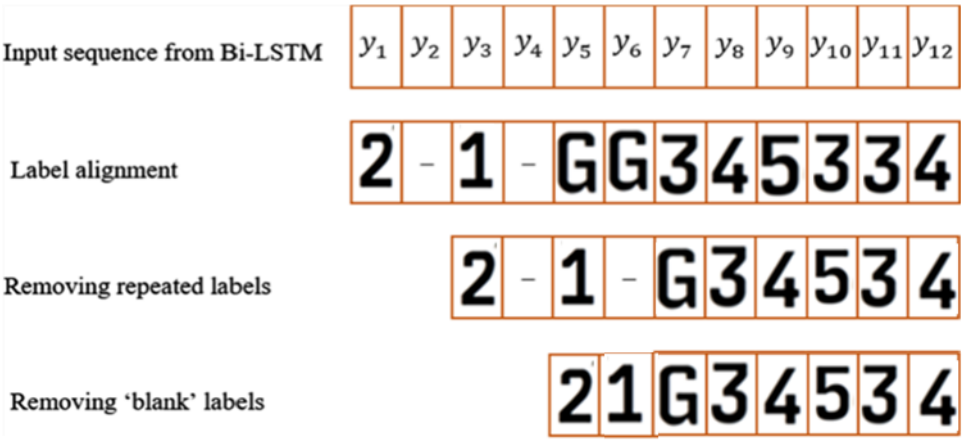


Fig. 2.15 The CTC computation diagram [43]

The Bi-LSTM input sequence appears in the first line. The label alignment procedure is shown in the second line. The final step is to eliminate redundant labels. Following the removal of "blank" labels, the final anticipated results are shown last.

2.7 Optical Character Recognition (OCR)

OCR is a technology that converts written or printed text in images into editable and searchable text. The OCR technology is integral to image processing and artificial intelligence [48]. The procedure for performing an OCR operation is summarized as follows:

- **Getting Images**: the first step is to prepare images that contain text, where they can be scanned papers, snapshots, or any other type of picture that includes visible text.
- **Preprocessing**: different processes are applied to improve the quality of the acquired images before proceeding with the text detection. This step could entail skew correction, binarization (turning the image into black and white), and noise reduction.
- **Text Detection**: this process locates the areas of the image that have text by separating the text from other items that are not text.

- **Character Segmentation**: in this step, the text is divided into individual characters. Accurately identifying each character depends on how accurate this step is.
- **Feature Extraction**: OCR algorithms examine every segmented character's characteristic, extracting crucial information that sets one character apart from another. Aspects like shape, size, and spatial relationships could be included in these traits.
- **Character Recognition**: every character is identified using the attributes that were extracted. To match the retrieved features with predetermined character patterns, OCR algorithms use machine learning models, neural networks, or pattern recognition approaches [49].

Chapter Three: Methodology

3.1 Introduction

In this chapter, the methodology of the traffic violations detection system is defined, and the methods used to create the system are also described. It offers an overview of the techniques used to create a solution to automatically detect traffic violations and identify the violated vehicle plate, which consists of several parts. First, background subtraction is used to identify vehicles, YOLOv4 is used to identify LPs, OCR is used to identify violated vehicle plates, and Twillo is a platform that helps developers around the world create unique communication features and capabilities such as voice, messaging, video, and email in their applications. In our proposed work, it is used to send a message to the owner of the violated vehicle.

3.2 The proposed System

The proposed traffic violation detection system uses a camera canon responsible for wire communication with an MSI computer (laptop) to indicate whether a violated vehicle is present. The Python program is based on CNN algorithms to detect and identify the plates of violated vehicles. After identifying the vehicle plate, it is sent to the Traffic Directorate to match it with the database for issuing a fine, along with sending a text message to the owner of the violated vehicle so that he can pay the fine. A flowchart of the proposed traffic violations detection system is shown in Fig 3.1.

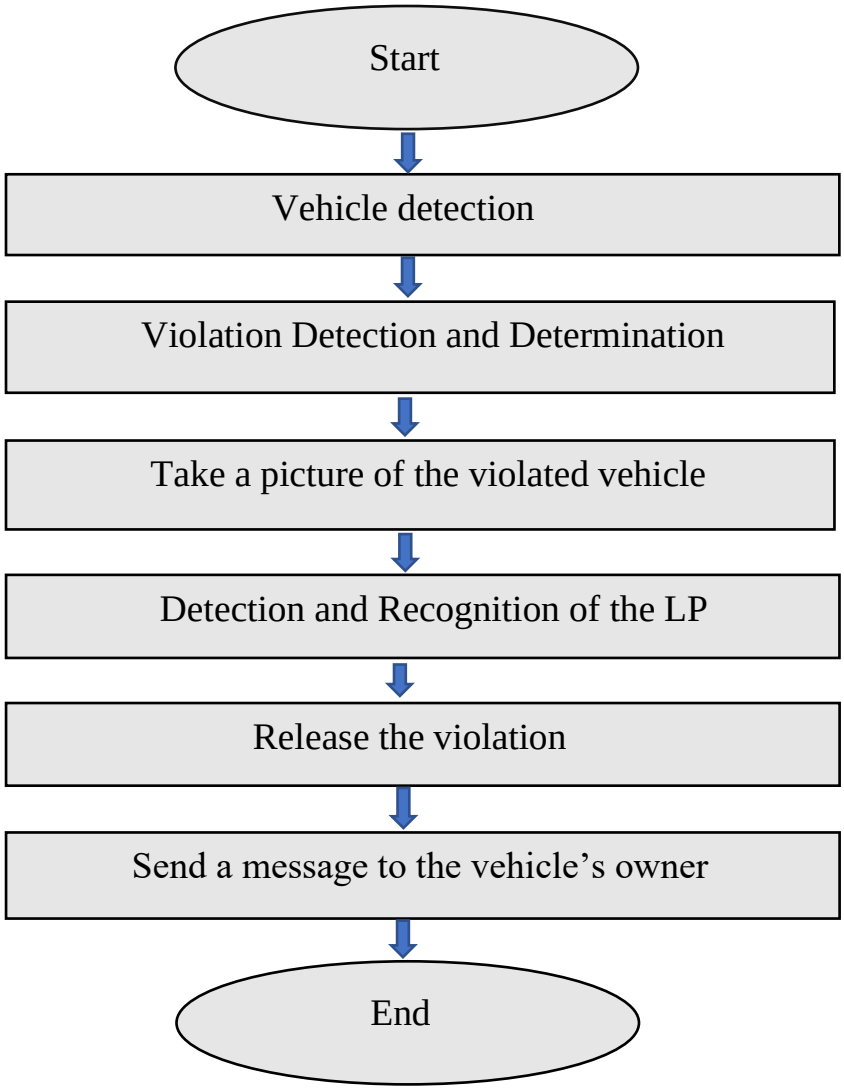


Fig. 3.1 Proposed Traffic Violations Detection System

3.3 System Initialization

At this point, the suggested system has been installed and configured. A camera has been installed on the top of a pedestrian bridge located in Karbala, in the middle of Ramadan street to obtain a clear view of the road, as shown in Fig. 3.2. In addition to drawing the count and speed lines, the distance threshold between them has also been established. Other thresholds

have been set up, such as the minimum vehicle size and the image threshold. A camera captures a data stream from the roads and transmits it to the system for further processing.

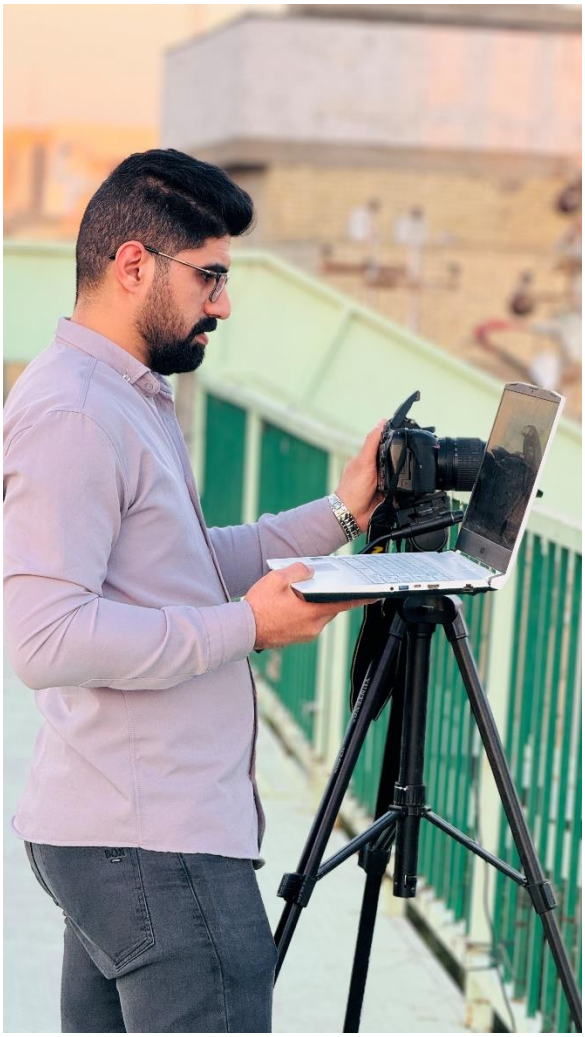


Fig. 3.2 Installing the camera on the pedestrian bridge

3.3.1 Vehicle Detection

The methods of background difference subtraction and color information contrast were employed for identifying the moving target vehicles. Initially, a video of vehicles moving on the street was recorded. Subsequently, the background subtraction approach was utilized to extract the moving and non-moving portions of the image, with the non-moving portion

regarded as the background. Overlaying the basic color information of the non-moving zone onto the moving region in the image enabled the creation of an approximate backdrop image, as the road information of the moving area shares similar features. Finally, the moving vehicle was identified using background difference subtraction.

3.3.1.1 Background Subtraction (BS)

The BS technique is used in object segmentation, security enhancement, pedestrian tracking, counting the number of visitors and vehicles in traffic, etc. It can learn and identify the foreground mask by extracting the moving foreground from the static background, as shown in Fig. 3.3. This work provides an algorithm that models the spatial interactions between pixels to promote a boundary that is compatible with space.

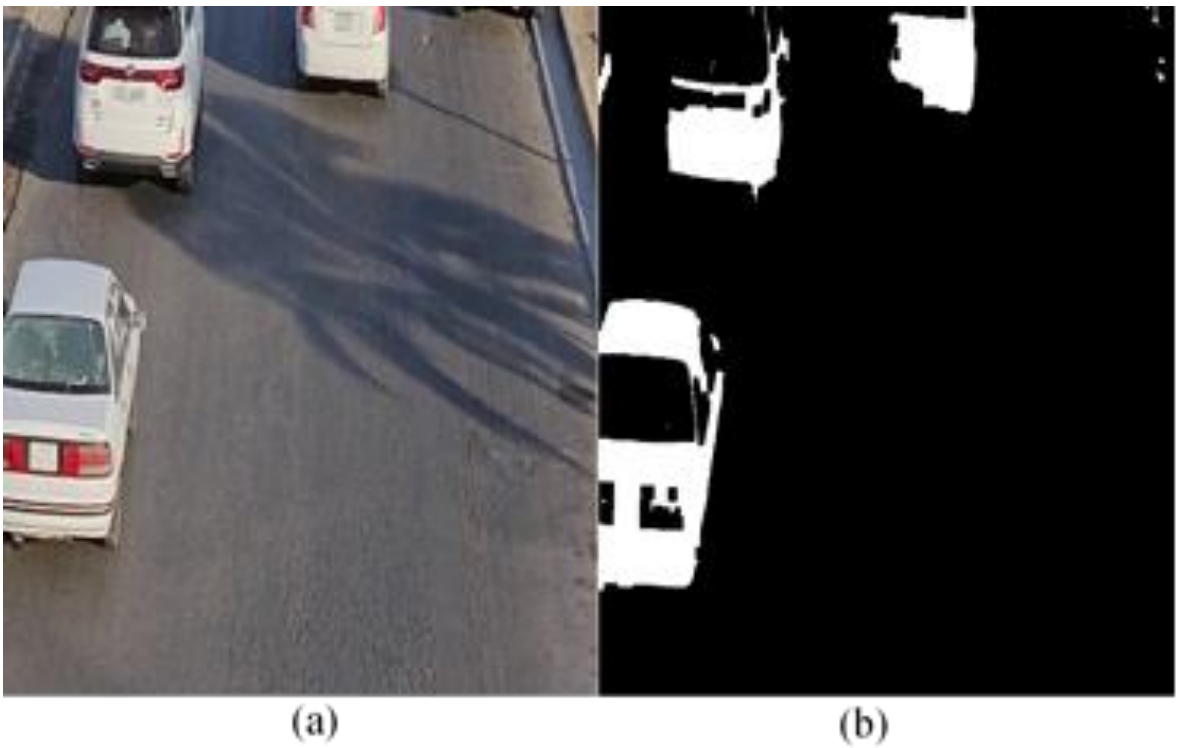


Fig. 3.3 Background Subtraction (a) Original image, (b) Background Subtraction result

The Mahalanobis distance (d_m) between the pixel and the background model is calculated as in Eq. **(3.1**) [50].

$$d_m = (x - \mu)^T \Sigma^{-1} (x - \mu) \quad \textbf{(3.1)}$$

where x is the color vector or intensity values in the video's current frame, μ is the mean vector of the background model that represents the average color or intensity values of the pixel over a series of frames, and Σ^{-1} is the inverse of the covariance matrix of the background model for each pixel location. The covariance matrix captures the variability and correlation of the color or intensity values of the pixel over the background frames. A threshold value is chosen to determine whether the object is a vehicle. If the distance is above the threshold, the pixel is considered a foreground; otherwise, it is regarded as a background. Therefore, the new pixel intensity is set to the pixel's value if it exceeds the threshold; otherwise, the pixels are set to 0, as shown in Fig. 3.4.

Fig. 3.4 Detection mask

3.3.1.2 Morphology

Morphology in the context of video processing refers to the application of morphological operations to a sequence of frames in a video. Just as in image processing, morphological operations in video are used for analyzing and manipulating the shapes and structures of objects within each frame of the video. Video morphology is particularly useful in computer vision applications, object tracking, and video analysis.

A. Erosion

Erosion at a specific pixel position is calculated by finding the minimum pixel value within the neighborhood defined by the structuring element, which is a matrix that defines the neighborhood used to process each pixel in the image. This operation has the effect of shrinking or eroding the boundaries of objects in the image. It is commonly used for noise reduction and separating connected objects in binary images.

B. Dilation

Dilation at a specific pixel position is calculated by finding the maximum pixel value within the neighborhood defined by the structuring element. This operation expands or dilates the boundaries of objects in the image. For each pixel in the binary image, the structuring element is centered on the pixel, and if any pixel within the structuring element is "1" (foreground), the central pixel in the output image is set to "1". This process results in larger objects, fills small holes, and connects close objects, enhancing features for better analysis in binary images. Dilation is commonly used in image preprocessing to improve object detection and shape analysis.

C. Mask1 opening morphology

The opening operation is an operation that combines erosion and dilation on each frame of a video sequence as shown in Fig. 3.5. This

operation is beneficial for removing small objects and noise while preserving the overall structure, where it was used for:

- **Noise Reduction:** the opening operation is frequently utilized to reduce noise and unwanted small elements in video frames. Its purpose is to smoothen the edges of objects and eliminate imperfections resulting in a depiction of objects.
- **Object Separation**: the opening technique can be applied to separate positioned or overlapping objects within video frames. Doing so facilitates the creation of boundaries between distinct objects making their analysis and tracking more manageable.
- **Preprocessing for Object Detection:** opening is commonly employed as a step before conducting object detection or recognition tasks. Its primary goal is to eliminate details while enhancing the visibility of more significant structures.
- **Enhancing Object Tracking:** while tracking objects across video frames, utilizing the opening operation can assist in maintaining consistency and continuity in object shapes.

Fig. 3.5 Applying mask1

D. Mask2 closing morphology

Closing operation is the application of dilation followed by erosion, as shown in Fig. 3.6. It is beneficial for closing small gaps and holes in objects while preserving the overall structure, where it is used for :

- **Object Segmentation:** the closing operation is used to improve the segmentation of objects in the video. It helps fill small gaps or holes in moving objects, making it easier to identify and track moving objects across frames.
- **Noise Reduction:** closing reduces noise and small unwanted details in video frames. It smoothens object boundaries and removes minor artifacts, contributing to a cleaner representation of objects.
- **Improving Object Tracking:** when tracking objects across video frames, the closing operation can help maintain the continuity and consistency of object shapes. This is very important when dealing with vehicles whose color is close to the background color.

Fig. 3.6 Applying mask 2

3.3.2 Violations Detection and Determination

The proposed system is designed to detect three traffic violations: exceeding the speed limit, driving in the opposite direction, and non-compliance with the Traffic Signal.

3.3.2.1 Exceeding the speed limit

As the process is accomplished using the recorded video via a camera, the speed is computed by sketching two lines on the street's image, as shown in Fig. 3.7.

Fig. 3.7 Drawing two lines on the street image

When the vehicle crosses the first line, the time at that moment, which is denoted as (T_0), is recorded. Also, when the vehicle reaches the second line, the (T_1) time is recorded. The time difference (T_d), which is spent between the two lines, is calculated as in Eq. (3.2).

$$T_d = T_1 - T_0 \tag{3.2}$$

The vehicle speed (s) is calculated as in Eq. (3.3).

$$s = \frac{d}{T_d} \tag{3.3}$$

where d is the distance between the two lines. The vehicle will be considered in violation if it exceeds the speed limit.

3.3.2.2 Driving in the opposite direction

The driving in the opposite direction can be detected by calculating the vehicle speed. As in the exceeding the speed limit violation, Eq. **(3.3)** is used to calculate the vehicle speed. When the speed appears to retain a negative value, this means that the vehicle is moving in the opposite direction, where it crosses the second line first and then touches the first line, as shown in Fig. 3.8.

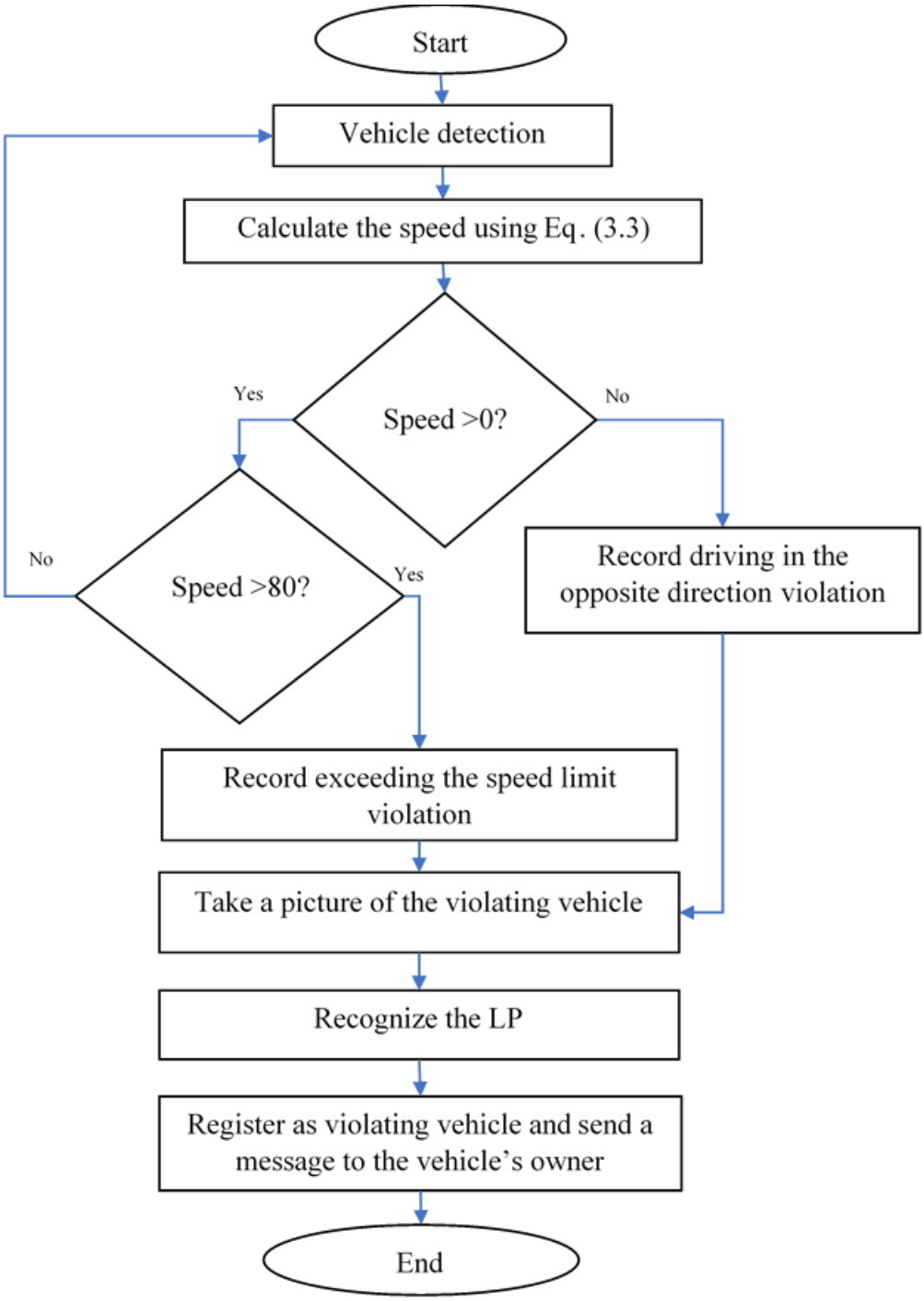


Fig. 3.8 Flowchart of the exceeding the speed limit and driving in the opposite direction violations

3.3.2.3 Non-compliance with the Traffic Signal

The system can detect vehicles that do not obey traffic signals if they cross the stopping line when the signal is red, as shown in Fig. 3.9.

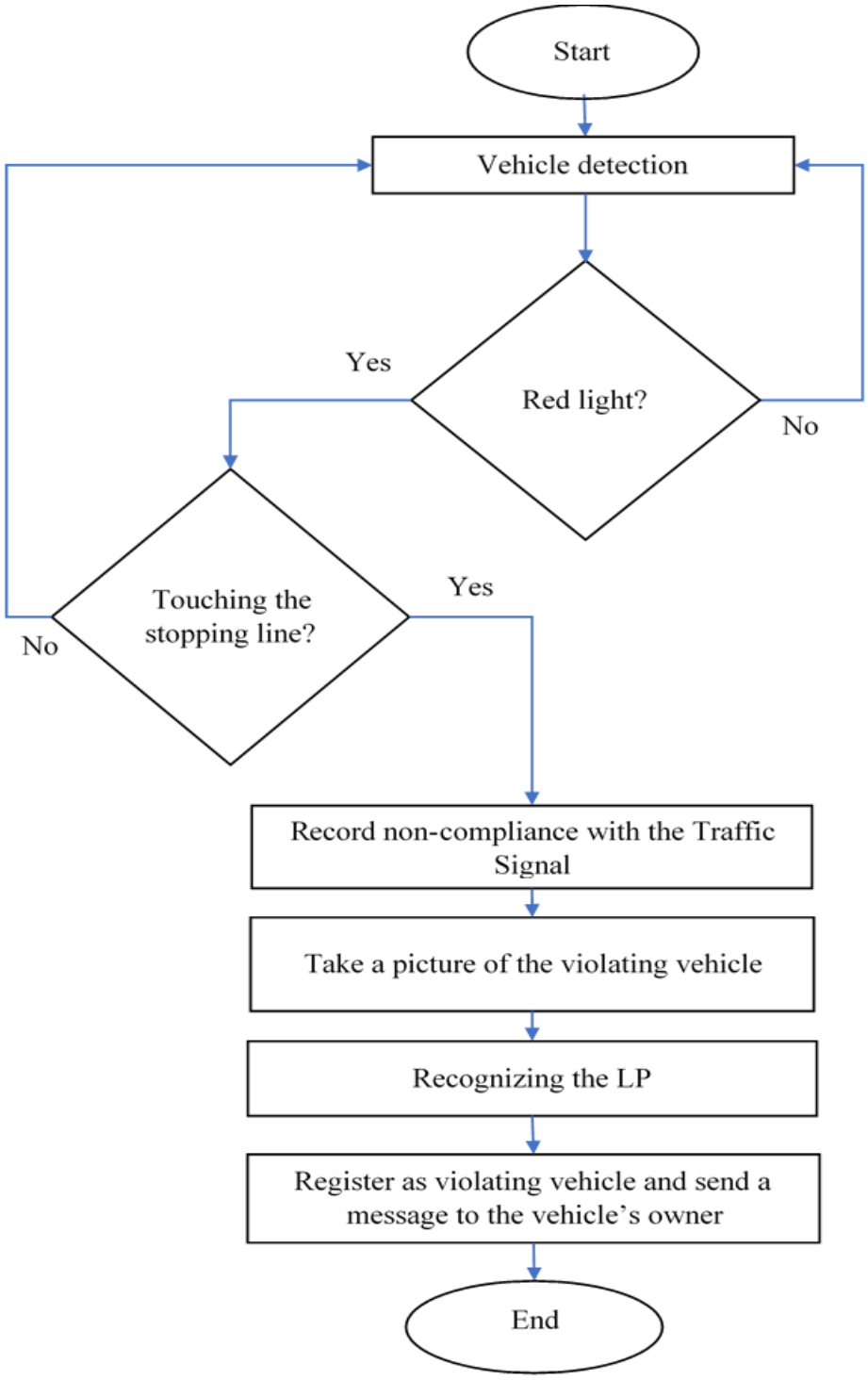


Fig. 3.9 Flowchart of the traffic light violation

3.3.3 Taking a Picture of the Violated Vehicle

A picture of the vehicle is taken after confirming the presence of a violation, where the picture is captured to the vehicle inside the square drawn around it as shown in Fig. 3.10. The vehicle is detected using the background subtraction technique.

(a) (b)

Fig. 3.10 Detecting a violated vehicle (a) Drawing a blue box and displaying the speed (b) Image of the violated vehicle

3.3.4 Detection and Recognition of the LP

After taking a photo of the violated vehicle, the work is done to detect the vehicle's LP using the YOLOv4 technology, and the LP is recognized using OCR. Two types of LPs are most used in Iraq: the first type contains Indian numbers and Arabic words, and the second one contains Arabic numbers and an English letter. For evidence, the LPs are distinguished based on their words or letters included. Therefore, the first type may called the Arabic type and the second one may called the English type.

3.3.4.1 YOLO Detection

The YOLO algorithm is used for LPs detection in conjunction with a CNN model. As shown in Fig. 3.11, it divides the input image into 18×18 grid cells. Every one of the grid cells predicts the object alone. There is a fixed number of boundary boxes predicted for each grid cell [21]. Predicting the five bounding box coordinates (c_x, c_y, w, h, c) is the responsibility of grid cell housing the item in its center. The object center in relation to the grid cell location is represented by the coordinates (c_x, c_y), while the width and height of the object in relation to image dimensions are represented by the coordinates (w, h). An object's presence in a grid cell is indicated by its confidence score (c) [41].

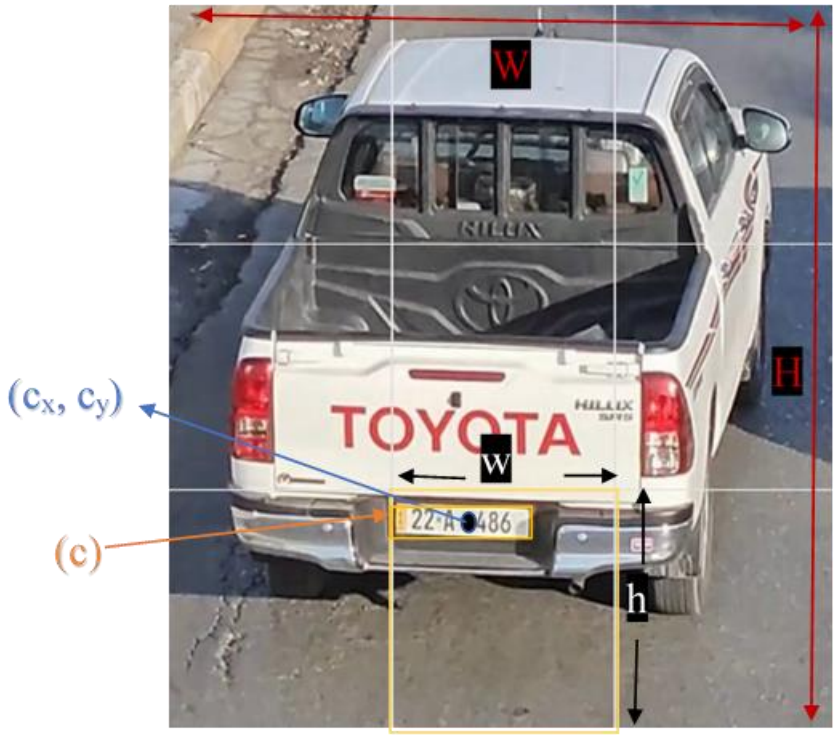


Fig. 3.11 LP Detection Using YOLO

3.3.4.2 Splitting the LP

Different types of LPs are used in Iraq, but two forms are the most widely used, as displayed in Fig. 3.12. The first form (Fig. 3.12 a) comprises three parts. The first part includes Hindi numbers and a letter that uniquely identifies each vehicle within the same type and registering province. The second part displays the name of the governorate where the vehicle was registered and the type of vehicle (Cargo, Taxi, Personal, Government, and Defense). The third part, which contains the word "IRAQ," is painted with some colors to classify the type of vehicle. The second form (Fig. 3.12 b) comprises a two-digit number representing the governorate in which the vehicle is registered, a letter, and a unique series of digit numbers for identification. On the left side of the plate, the same colors of the first form are used.

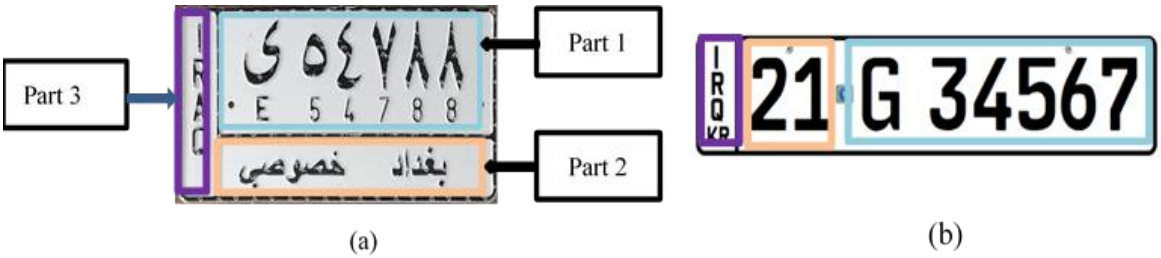


Fig. 3.12 Types of Iraqi vehicle LPs (a) First form and (b) Second form

3.3.4.3 Identifying the LP Type

The average confidence score for the two languages, English (En) and Arabic (Ar), is calculated based on the confidence scores obtained from the OCR results to identify the LP type. Fig. 3.13 shows the flowchart of identifying the LP type. The following steps illustrate the process of calculating the confidence scores:

- Initialize confidence variables: two variables, *EnConf* and *ArConf*, are initialized to zero. These variables will hold the average confidence scores for English and Arabic, respectively.
- Check for OCR results: OCR results are checked for each language, which is done by verifying if the length of the list of confidence scores (*EnScore* for English and *ArScore* for Arabic) is greater than 0. If there are no OCR results for a language, the confidence score for that language remains zero.
- Calculate average confidence scores: if there are OCR results for English ($EnConf > 0$), calculate the average confidence score for English ($EnConf$) by dividing the sum of all the confidence scores in the EnScore list by the total number of confidence scores (n). Equation **(3.4**) is used to obtain the *EnConf*.

$$EnConf = \frac{\sum EnScore}{n} \quad \textbf{(3.4)}$$

Similarly, if there are OCR results for Arabic (*ArScore* > 0), calculate the average confidence score for Arabic (*ArConf*) by dividing the sum of all the confidence scores in the ArScore list by the total number of confidence scores (m), as in Eq. **(3.5**).

$$ArConf = \frac{\sum ArScore}{m} \quad \textbf{(3.5)}$$

The average confidence score calculations (*EnConf* and *ArConf*) include each language's confidence level in the OCR results. The OCR results are first checked to avoid divide-by-zero errors. If there are scores, it calculates the average confidence score by summing the individual confidence scores and dividing them by the number of scores. These average confidence

scores are then used to make decisions about the language detected and the LP type chosen.

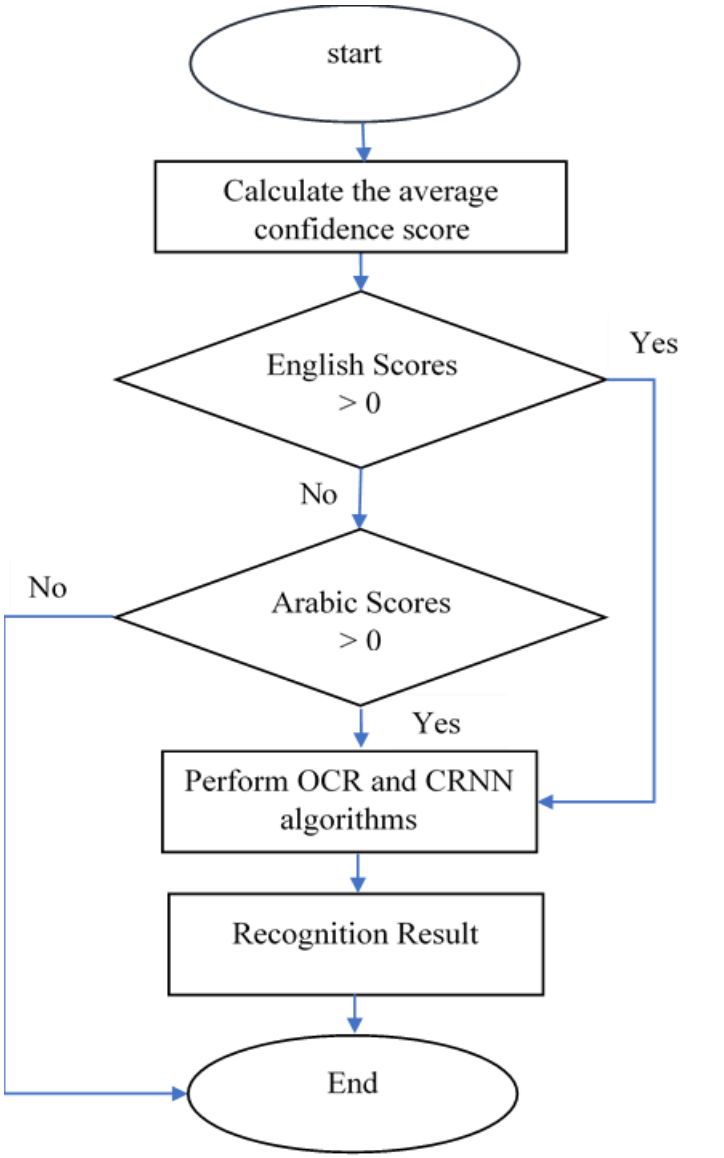


Fig. 3.13 Flowchart of the LP recognition process using OCR and CRNN

3.3.4.4 Color Detection

As previously mentioned, the color of the rectangular portion on the LP left side corresponds to the type of vehicle. The color is determined through specific calculations, as demonstrated below. In practice, the ratio of the colored part width to the total width of the LP is about (0.0738255). Therefore,

the width of the colored part in the detected LP is estimated by multiplying the total width of the detected LP by 0.0738255. The average color is calculated as illustrated in Fig. 3.14 to eliminate the effects of light illumination, dust, and some painting removal.

A threshold value is applied to the resulting average color, which is about 200 pixels. If the value of the average color is more than 200, it is set to 255, but if it is less than 200, it is reset to 0. Table 3.1 displays the information regarding the color in the LP.

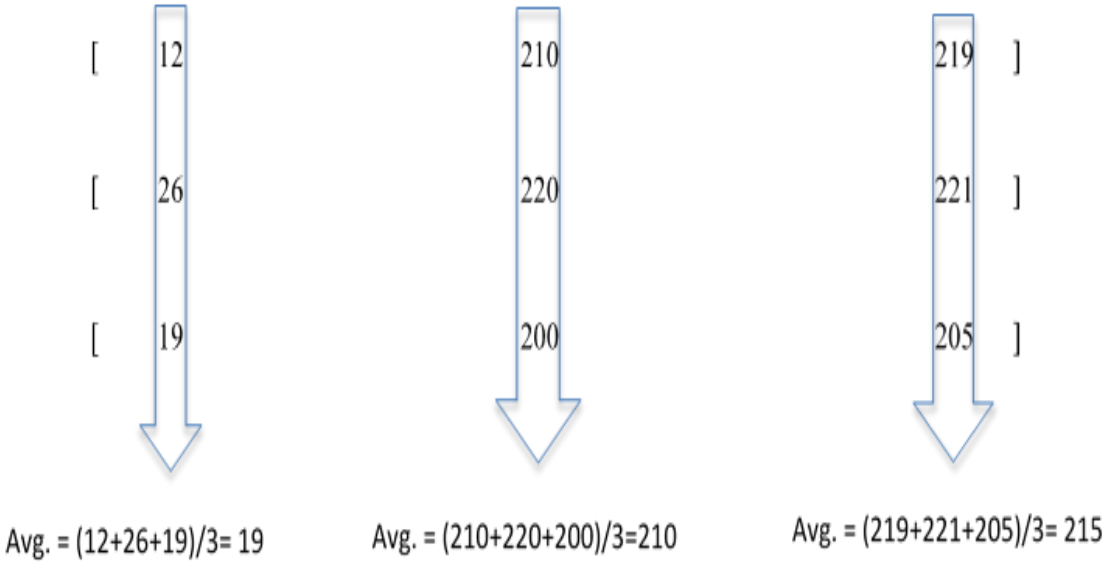
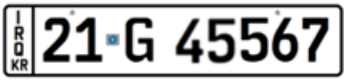
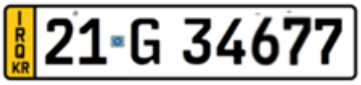
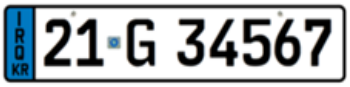
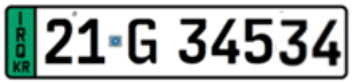


Fig. 3.14 Example of calculating average pixel values for the colored region

Table 3.1 Information regarding the colored region

Color	Average pixel values	Vehicle Type	LP Type
White	[255, 255, 255]	Personal	
Red	[0, 0, 255]	Taxi	

Yellow	[0, 255, 255]	Cargo	
Blue	[255, 0, 0]	Government	
Green	[0, 255, 0]	Defense	

The computed average color is compared with the standard average pixel values for each color type. For the example shown in **Fig. 3.14**, the average color after applying the threshold value becomes [0 255 255], which refers to the Yellow color .

3.3.4.5 Characters recognition

LPs are recognized using OCR with the CRNN algorithm. The network architecture of CRNN, as shown in Fig. 3.15, consists of three components, including the convolutional layers, the recurrent layers, and a transcription layer, from bottom to top. At the bottom of CRNN, the convolutional layers automatically extract a feature sequence from each input image. On top of the convolutional network, a recurrent network is built to predict each feature sequence frame outputted by the convolutional layers. The transcription layer at the top of CRNN is adapted to translate the per-frame predictions by the recurrent layers into a label sequence.

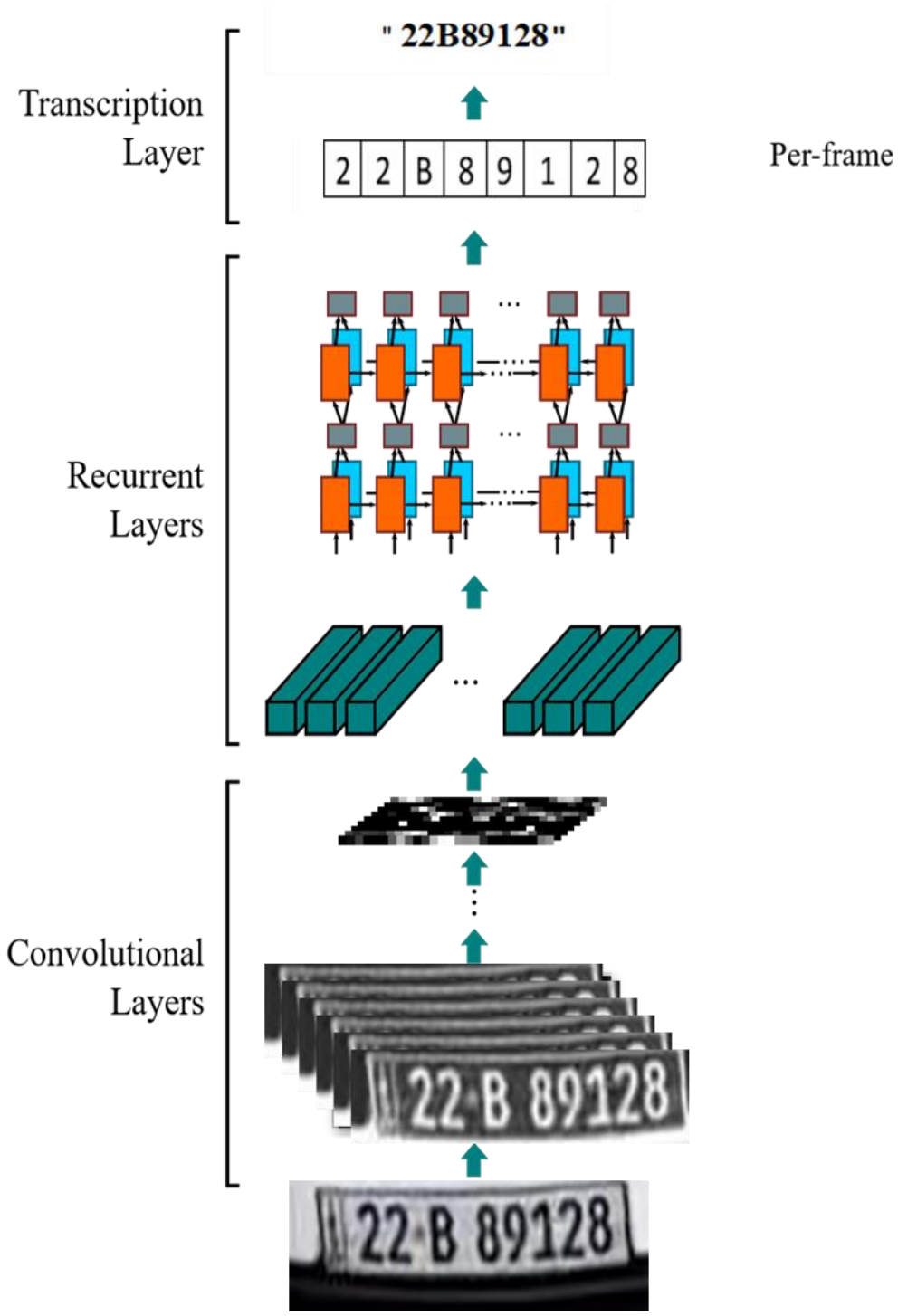


Fig. 3.15 The CRNN layers used in LP recognition

3.3.5 Releasing the violation

After recognizing the LP, a violation is issued to the vehicle owner through the Twilio company, which supports multiple programming

languages such as Python, JavaScript and Java. Once a violation is registered on the Twilio platform, a text message is sent to the General Directorate of Traffic to register the violation, issue the fine, and send a message to the vehicle owner to inform him of the violation. This message includes all the details regarding the type of violation, the date and time of the violation, and the location of the violation. In addition, the most crucial element in this message is the vehicle number that was previously recognized, as shown in Fig. 3.16.

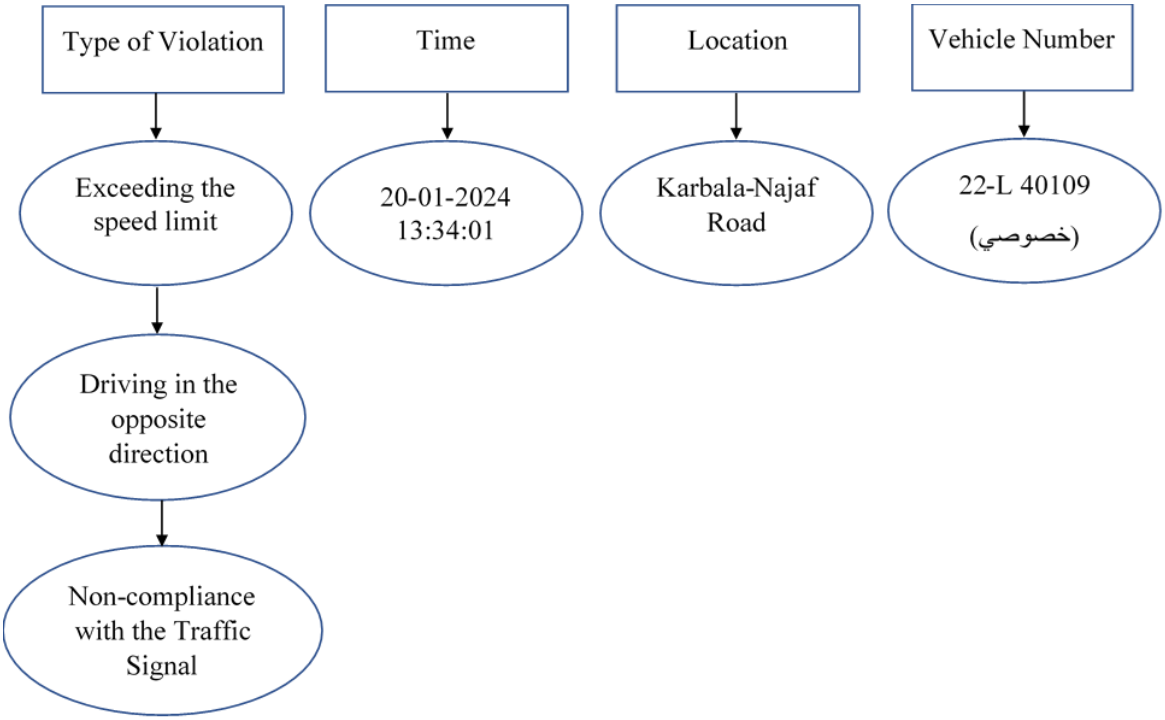


Fig. 3.16 Message Components

Chapter Four: Experimental Results and Discussion

4.1 Introduction

This chapter presents the results of detecting traffic violations and proposes three types of violations: Exceeding the speed limit, Driving in the opposite direction and Non-compliance with the Traffic Signal. In addition, detection and recognition of LPs using YOLOv4 and CRNN algorithms with OCR, respectively are also done in this chapter. The results discuss several stages, including detecting the violation, its type, and LP recognition.

4.2 Experimental results

The PYTHON 3.12.4 programming language is used to validate the developed algorithm. A personal computer is used for simulation with the following specifications: Intel Core i7 processor with a speed of 2.3 GHz and 16 GB Memory, and Windows 11 operating system. Canon camera is used with the following specifications: IR, Optimised Infrared Filter, 30.3 Megapixels, Full frame CMOS sensor, -6 EV AF, Focus in low light, Vari-angle LCD, 30x Magnification and 4K and 10-bit movies.

4.2.1 Experiments on Detecting Traffic Violations

In this test, the system is used to monitor traffic violations using a high-resolution camera. It is capable of detecting three types of violations:

- Non-compliance with the traffic signal
- Exceeding the speed limit
- Driving in the opposite direction

If a vehicle has committed multiple violations, it will be penalized accordingly. The system uses a background subtraction algorithm to identify vehicles, and their speeds are calculated according to Eq. (3.3). The YOLOv4 algorithm is used to determine the vehicle's LP, and the recognition of the LP is done using OCR. The detection rates of the three violation types are displayed in **Table 4.1** with the LP recognition rates.

Table 4.1 Experimental results for violation detection and recognition of Iraqi LPs

Violation type	No. of violated vehicles	No. of vehicles detected	Violation Detection rate (%)	Recognition rate of LPs (%)
Non-compliance with the Traffic Signal	105	105	100	99.02
Exceeding the speed limit	72	69	95.83	97.11
Driving in the opposite direction	30	29	96.67	98.53
All violations	207	203	98.06	98.22

Multiple videos are filmed at different times of the day to detect traffic violations. For exceeding the speed limit violation, 72 vehicles that exceed the limit are tested, and 69 are correctly detected. Besides, the accuracy rate of recognizing the LPs is 97.11%. When testing the detection of driving in the opposite direction violation for 30 vehicles, 29 vehicles are detected, and the LP recognition rate is 98.53%. During the test, all violated vehicles that do not respect a red signal are detected.

4.2.1.1 Vehicle Detection

The system relies primarily on detecting moving vehicles to determine whether it violates traffic rules.

Moving vehicles are detected using background subtraction by specifying the minimum vehicle size that is previously specified. If the size of the detected object is greater than the minimum vehicle size, it is detected as a vehicle, and a box is drawn around the detected object, (as shown in Fig. 4.1). It is neglected if the size of the detected object is less than the minimum size of the vehicle. The system achieves an excellent accuracy rate in detecting moving vehicles with an accuracy of 99%.

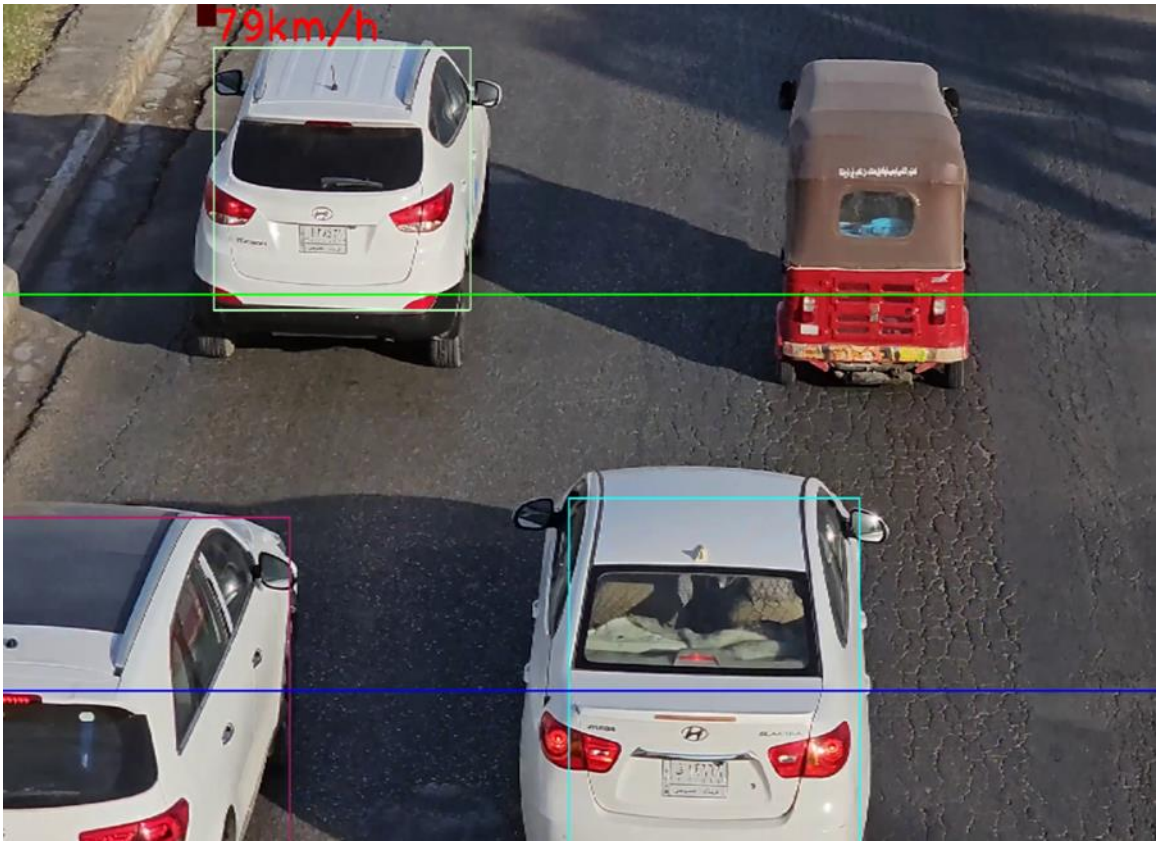


Fig. 4.1 Vehicle Detection

4.2.1.2 Non-compliance with the traffic signal

The first violation to be checked is Non-compliance with the Traffic Signal where the system checks the vehicle if it crosses the first line, as shown in Fig. 4.2. The system achieves a high accuracy rate in detecting this violation

with 100% accuracy, demonstrating an excellent recognition rate accuracy for the LPs of violated vehicles at 99.02%.

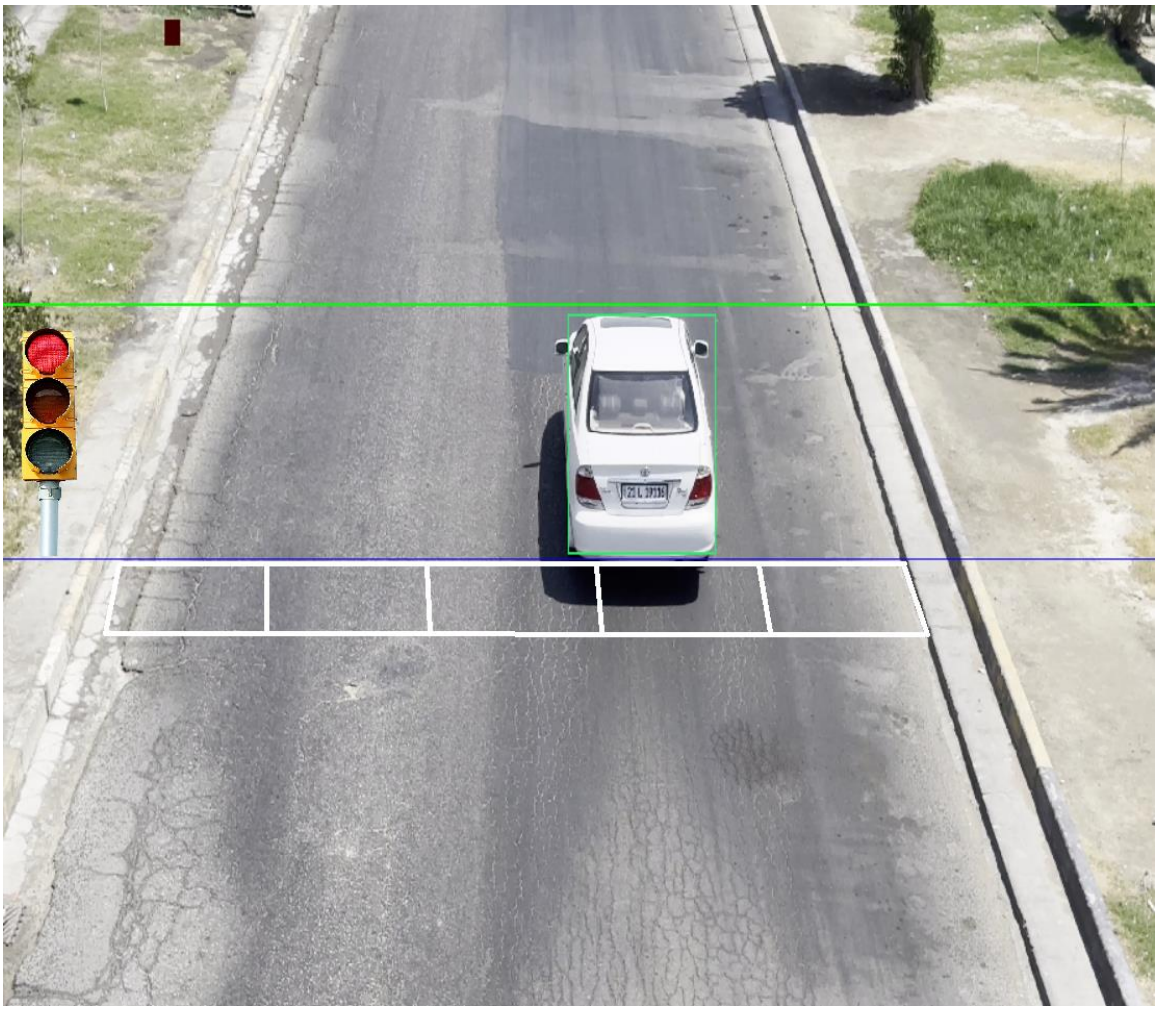


Fig. 4.2 The Vehicle Touches the Stopping Line

4.2.1.3 Exceeding the speed limit

The system showed excellent results in detecting speed violation according to the time and distance calculations. The camera's location is taken into account at a height of five meters from the street, and the system achieved an accuracy of 95.83% in detecting speeding vehicles.

Fig. 4.3 shows the vehicle that crossed the two lines and its speed is measured by dividing the fixed distance between the two lines over the time

taken for the vehicle to cross from the first line, in blue, to the second line, in green. With this test, the recognition rate for the LPs is excellent with an accuracy of 97.11%.

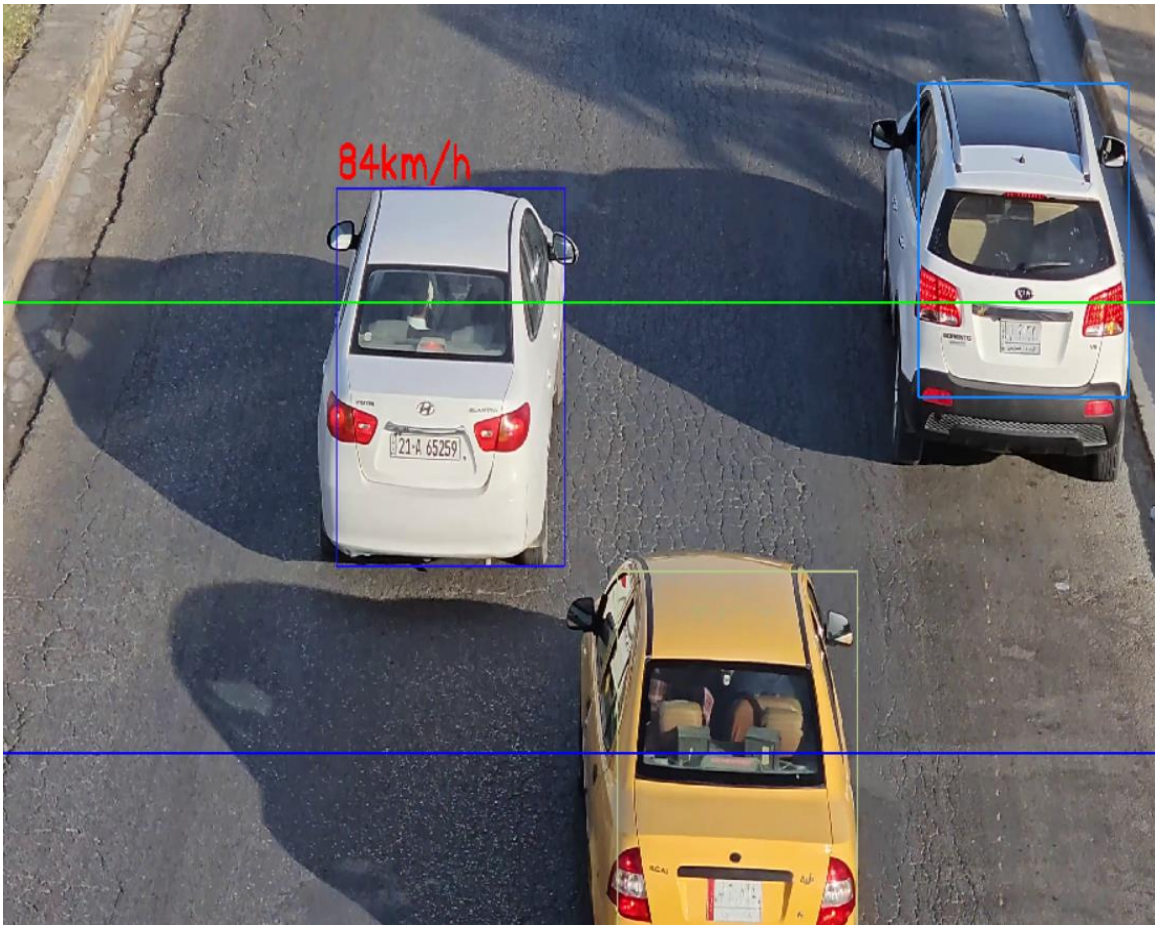


Fig. 4.3 Measuring the vehicle's speed after touching the second line

4.2.1.4 Driving in the opposite direction

The proposed system achieves good accuracy of 96.67% in detecting vehicles moving in the opposite direction, and excellent recognition rate of LPs with an accuracy of 98.53%. Fig. 4.4 shows a vehicle moving in the opposite direction, which shows that the vehicle's speed appears to be a negative value.

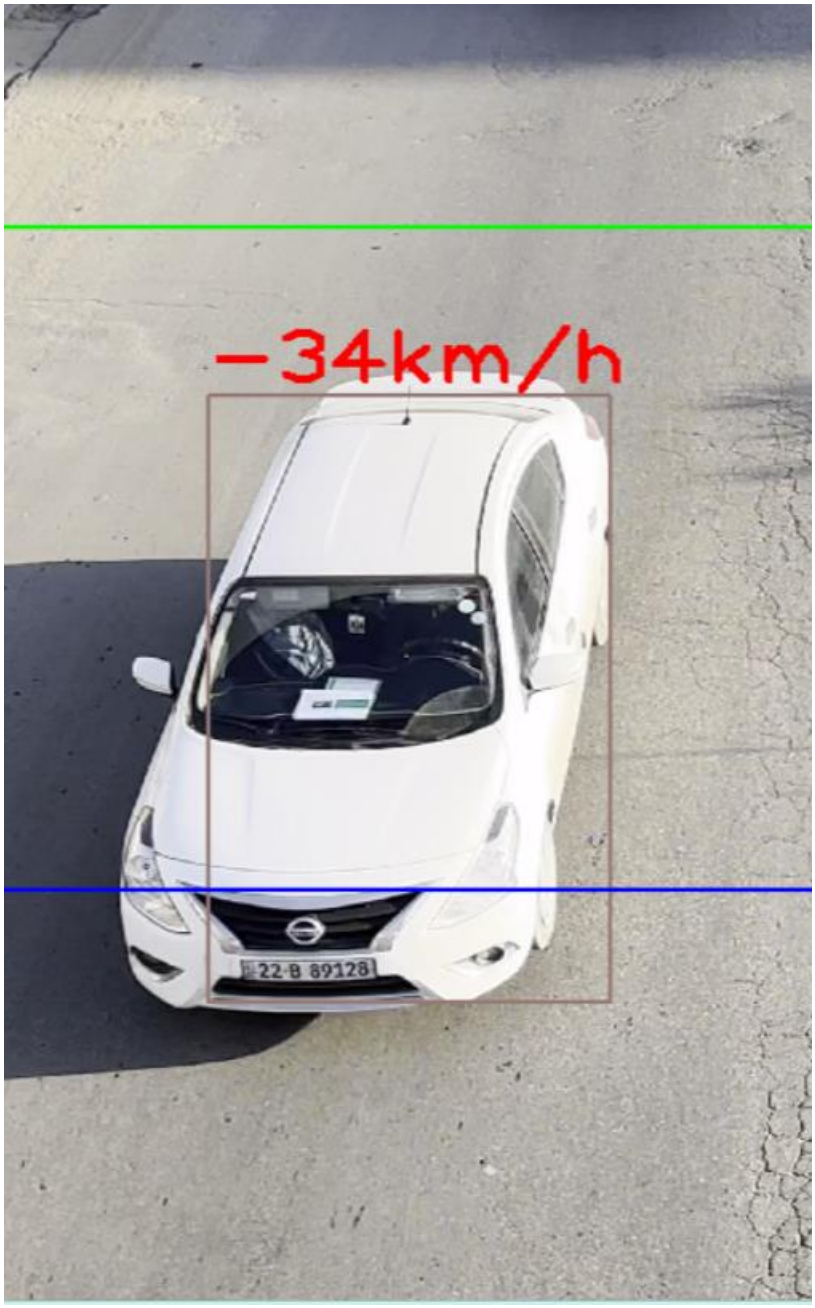


Fig. 4.4 Vehicle moving in the opposite direction

4.2.2 Experiments on LP Recognition

4.2.2.1 Experiment 1

Data collection is a crucial phase in recognition systems. The proposed work utilizes Google's Open Image dataset to train the YOLOv4 detector, which includes thousands of object photographs annotated for object detection and segmentation [51]. The portion of the dataset that deals with LPs comprises 1500 training images and 300 testing images in the YOLO format, collected with more explanation on the usage in [51]. Fig. 4.5 provides samples of mentioned dataset.

Fig. 4.5 Samples of open images dataset [51]

This experiment demonstrates the system's performance in identifying LPs using the open image dataset comprising 1800 vehicle images in various lighting conditions. The results are presented in Table 4.2.

Table 4.2 Experimental results of using the open image dataset

No. of training images	No. of testing images	Testing Rate (%)
1500	300	90

4.2.2.2 Experiment 2

Several images taken from a video recorded with a high-resolution camera at a rate of 30 frames per second are tested. As mentioned previously, after detecting the violated vehicle, a photo of the vehicle is taken according to the box drawn around it to determine the LP and identify it. The results of detecting and recognizing the LP are presented in Table 4.3.

Table 4.3 Experimental results of the second type of Iraqi LP detection and recognition for moving vehicles

No. of vehicle Images	No. of detected LP	No. of recognized LP	Detection Rate (%)	Recognition Rate (%)
115	114	113	99.13%	98.2%

The proposed LPR system evaluates Iraqi LPs using variable-sized colored images. Table 4.4 shows the achieved results using our own database.

Table 4.4 Experimental results of LPs detection and recognition regarding the two forms of Iraqi LPs

LP Type	No. of images	Detection rate (%)	Recognition rate (%)
Arabic LP	40	97.5	94.87
English LP	40	100	97.5

As illustrated in Table 4.4, all the images of the second form are detected, whereas one image of the first form is undetected. The main reasons of not correctly extracting an image are that it may have some blur effects or the process of cleaning up the unwanted objects discards out some critical

information, and that caused the incorrect crop of the LP, as shown in Fig. 4.6 for a sample.

Fig. 4.6 Sample of incorrectly detected LP

For the recognition step, only one LP of the second form and two LPs of the first form are not identified. Fig. 4.7 shows the result of detecting and recognizing some LPs.

4.2.2.3 LP Recognition Speed

Usually, the speed of the various optical recognition methods depends mainly on the accuracy of input images because it specifies the number of pixels required for processing, where this issue is overcome with the proposed system. Because grids reduce all input images to 416 × 416 before inserting them into the first layer, speed is nearly invariant to input image size. Table 4.5 shows the time required for detecting and recognizing the LP.

Table 4.5 Processing speed of detecting and recognizing the LP

Stage	Minimum time (s)	Maximum time (s)	Average time (s)
Detection	0.18	0.21	0.195
Recognition	1.0	1.4	1.2

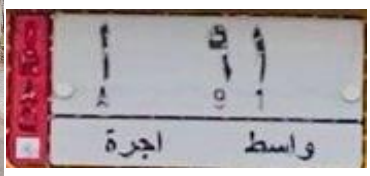
Vehicle image	Detected LP	Recognition result
		img=A10.jpg, text=أ٥٥٥١٩ خصوصي القادسية', isArabic=True
		img=A3.jpg, text= 22-H 73247 (خصوصي), isArabic=False
		img=img.png, text=١٩١ اجرة واسط , isArabic=True
		img=photo_8.jpg, text= 24-A 41280 (حمل), isArabic=False

Fig. 4.7 Results of LP recognition

4.3 Sending a Short Message Service (SMS)

The system aims to send a message to the general traffic department to issue a fine for the violated vehicle, in addition to sending a text message to the owner of the violated vehicle to know and pay the financial fine before a specified time where the fine is doubled. This message is sent using the Twilio application which contains the type, time, and place of the violation, as well as the vehicle's LP as shown in Fig. 4.8.

Fig. 4.8 SMS sent to the vehicle's owner

Fig. 4.9 shows the process of detecting different types of violations and the results of recognizing the LPs of the violated vehicles.

Fig. 4.9 Violation detection (a) Calculating vehicles' speed (b) Acquiring images of the violated vehicles (c) Detecting LPs, and (d) Recording the violations

4.4 Comparison with other work

It is crucial to compare the proposed work outcomes with those of the related work. As per the literature survey, there is no common dataset

employed in all or most approaches. However, each approach has used a dedicated dataset created by collecting some images. Therefore, a comparison is made in Table 4.6 with the related work despite the dataset used. It is worth noting that the system works to detect more than one violation at a time, in addition to using the camera only without using any sensor to detect violations.

Table 4.6 Comparison with the related work regarding the violation detection

Ref. No.	Year	Method	Violation Type	Accuracy rate (%)
[10]	2016	BLOB analysis and Kalman filter	Exceeding the speed limit	86.11
[11]	2020	Sensors	Exceeding the speed limit	90
[12]	2020	SORT	Exceeding the speed limit	78
[13]	2020	BLOB	Traffic signal	71
[14]	2021	PSO	Traffic signal	91.1
[15]	2021	YOLOv3	Exceeding the speed limit	89.24
[16]	2022	Depth and GAC	Traffic signal	81.9
[17]	2022	RF+GBDT	Traffic signal	96
[18]	2023	YOLO and HSA	Traffic signal	88.24
Proposed Work	2024	BS and YOLOv4	Exceeding the speed limit, Driving in the opposite direction, and Non-compliance with the Traffic Signal	98.06

As illustrated in Table 4.6, the proposed work achieves an accuracy rate of about 98.06%, outperforming other related work. Moreover, the proposed

system detects three violations simultaneously, unlike the others that can detect one type of violation.

In the case of recognizing the LPs, a comparison between the proposed work and the related work is made in Table 4.7.

Table 4.7 Comparison with the related work regarding the recognition of LPs

Ref. No.	Yeas	Method	Accuracy rate (%)
[19]	2018	BPNN	97.7
[20]	2019	TM	96
[11]	2020	OCR	56.67
[21]	2019	OCR	78
[22]	2020	CNN+OCR	89.15
[23]	2020	LSTM+CNN	85
[24]	2020	KNN	90
[25]	2021	CRNN	95
[26]	2021	Faster R-CNN	98.02
[27]	2023	CNN+YOLO	93
Proposed Work	2024	OCR + CRNN	98.53

According to the results presented in Table 4.7, the proposed work achieves a recognition rate of about 98.53%, which is the best among related work. In contrast to associated approaches, the LPs are detected and identified while the vehicles are in motion rather than stationary.

Chapter Five: Conclusion and Future Work

5.1 Conclusion

This work introduces a robust system for detecting traffic violations using video processing and deep learning. Background subtraction technology is used to detect moving vehicles, with morphological operations used to improve the performance of detecting moving vehicles by filling the blanks in the images that contain detected vehicles. LPs are identified using YOLOv4 technology trained on the open image dataset in advance, in addition to the OCR technology and the CRNN algorithm that has the ability to learn from time sequences, which helps in understanding the relationship between letters and numbers and recognizing the LPs correctly. Utilizing YOLOv4, CRNN, and OCR algorithms produces the best results for detecting and recognizing LPs compared to other methods. The proposed system demonstrates an applicable real-time approach for detecting traffic violations, achieving a 98.06% overall accuracy rate for offenses such as exceeding the speed limit that is calculated based on the time difference that the vehicle spends in a specific distance, driving in the opposite direction when the speed signal is kept negative, and exceeding the traffic signal that is detected through the red light signal. Instead of using sensors to detect violations and LPs, which led to a delay in the system's response, the system used only a camera, which increases the speed of detecting violations and LPs.

On the other hand, the proposed system has achieved excellent results in recognizing LPs, with a success rate of 94.87% for the first type containing Hindi numbers for a fixed vehicle. The system achieves a 98.22% accuracy rate for the second type, including Arabic numbers for moving vehicles. This

achievement is necessary to identify the violated vehicles and enforce relevant penal efforts accurately. The comparisons with the previous approaches show the superiority of the proposed work, which achieves the highest accuracy rate in violation detection and LPs recognition. With more than one violation detected at the same time, the violation is issued, and a message is sent to the vehicle owner.

5.2 Future Work

Due to the importance of this system in reducing traffic accidents and preserving people's lives. The following suggestions can be taken into account for future work:

1. Detecting violations such as not wearing a seat belt and using a mobile phone while driving.
2. Adding a violation for not wearing a helmet for drivers of two-wheeled motorcycles.
3. Improving the accuracy of the system in different weather conditions.

References

[1] A. C. P. Uy *et al.*, “Automated traffic violation apprehension system using genetic algorithm and artificial neural network,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 2094–2099, 2017, doi: 10.1109/TENCON.2016.7848395.

[2] F. Pan, Y. Yang, L. Zhang, C. Ma, J. Yang, and X. Zhang, “Analysis of the Impact of Traffic Violation Monitoring on the Vehicle Speeds of Urban Main Road: Taking China as an Example,” *J. Adv. Transp.*, vol. 2020, 2020, doi: 10.1155/2020/6304651.

[3] F. Chen, M. Song, and X. Ma, “Investigation on the injury severity of drivers in rear-end collisions between cars using a random parameters bivariate ordered probit model,” *Int. J. Environ. Res. Public Health*, vol. 16, no. 14, 2019, doi: 10.3390/ijerph16142632.

[4] H. Zheng, W. Chang, and J. Wu, “Traffic flow monitoring systems in smart cities: Coverage and distinguishability among vehicles,” *J. Parallel Distrib. Comput.*, vol. 127, pp. 224–237, 2019, doi: 10.1016/j.jpdc.2018.07.008.

[5] Z. Cheng, J. Lu, Z. Zu, and Y. Li, “Speeding Violation Type Prediction Based on Decision Tree Method: A Case Study in Wujiang, China,” *J. Adv. Transp.*, vol. 2019, 2019, doi: 10.1155/2019/8650845.

[6] A. T. Bhat, Anupama, Akshatha, M. S. Rao, and D. G. Pai, “Traffic violation detection in India using genetic algorithm,” *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 309–314, 2021, doi: 10.1016/j.gltp.2021.08.056.

[7] A. Alaydrusl, W. K. Putra, Y. Nugroho, Muhammad, and N. Surantha, “A review of traffic violation detection technology in reporting mechanism,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 729, no. 1, 2021,

doi: 10.1088/1755-1315/729/1/012005.

[8] R. Shreyas, B. V. P. Kumar, H. B. Adithya, B. Padmaja, and M. P. Sunil, "Dynamic traffic rule violation monitoring system using automatic number plate recognition with SMS feedback," *2nd Int. Conf. Telecommun. Networks, TEL-NET 2017*, vol. 2018-Janua, pp. 1–5, 2017, doi: 10.1109/TEL-NET.2017.8343528.

[9] M. A. Rafique, W. Pedrycz, and M. Jeon, "Vehicle license plate detection using region-based convolutional neural networks," *Soft Comput.*, vol. 22, no. 19, pp. 6429–6440, 2018, doi: 10.1007/s00500-017-2696-2.

[10] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim, and E. P. Dadios, "Intelligent system architecture for a vision-based contactless apprehension of traffic violations," in *2016 IEEE Region 10 Conference (TENCON)*, IEEE, 2016, pp. 1871–1874.

[11] J. Yim, R. A. Cadiente, G. P. Mayuga, and E. R. Magsino, "Integrated plate recognition and speed detection for intelligent transportation systems," in *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, IEEE, 2020, pp. 6–11.

[12] A. Grents, V. Varkentin, and N. Goryaev, "Determining vehicle speed based on video using convolutional neural network," *Transp. Res. Procedia*, vol. 50, pp. 192–200, 2020.

[13] M. M. Bachtiar, A. R. Mawardi, and A. R. A. Besari, "Vehicle classification and violation detection on traffic light area using BLOB and mean-shift tracking method," in *2020 International Conference on Applied Science and Technology (iCAST)*, IEEE, 2020, pp. 94–98.

[14] S. S. Wankhede and P. Bajaj, "Traffic Violation Detection Model Using Soft Computing Tools," *2021 6th Int. Conf. Converg. Technol. I2CT*

2021, no. April 2021, 2021, doi: 10.1109/I2CT51068.2021.9417887.

[15] S. R. Anand, N. Kilari, and D. U. S. R. Kumar, "Traffic signal violation detection using artificial intelligence and deep learning," *Int. J. Adv. Res. Eng. Technol.*, vol. 12, no. 2, pp. 207–217, 2021.

[16] Y. T. Peng, C. Y. Liu, H. H. Liao, W. C. Lien, and G. S. J. Hsu, "Traffic Violation Detection via Depth and Gradient Angle Change," *2022 IEEE 7th Int. Conf. Intell. Transp. Eng. ICITE 2022*, no. November 2022, pp. 326–330, 2022, doi: 10.1109/ICITE56321.2022.10101435.

[17] M. Zahid, A. Jamal, Y. Chen, T. Ahmed, and M. Ijaz, "Predicting red light running violation using machine learning classifiers," in *Green Connected Automated Transportation and Safety: Proceedings of the 11th International Conference on Green Intelligent Transportation Systems and Safety*, Springer, 2022, pp. 137–148.

[18] T. Singh, V. Rajput, Satakshi, U. Prasad, and M. Kumar, "Real-time traffic light violations using distributed streaming," *J. Supercomput.*, vol. 79, no. 7, pp. 7533–7559, 2023.

[19] F. Xie, M. Zhang, J. Zhao, J. Yang, Y. Liu, and X. Yuan, "A robust license plate detection and character recognition algorithm based on a combined feature extraction model and BPNN," *J. Adv. Transp.*, vol. 2018, 2018.

[20] A. A. J. Altameemi, H. H. Abbas, and A. Alabaichi, "Iraqi vehicle license plate recognition using template matching technique," *ICIC Express Lett. Part B Appl.*, vol. 10, no. 6, pp. 465–473, 2019, doi: 10.24507/icicelb.10.06.465.

[21] Hendry and R. C. Chen, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning," *Image Vis. Comput.*, vol. 87, pp. 47–56, 2019, doi: 10.1016/j.imavis.2019.04.007.

[22] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, p. 102773, 2020, doi: 10.1016/j.jvcir.2020.102773.

[23] A. Anson and T. Mathew, "Detection and Recognition of License Plate Using CNN and LSTM," in *Advances in Communication Systems and Networks: Select Proceedings of ComNet 2019*, Springer, 2020, pp. 701–721.

[24] D. A. Abd Alhamza and A. D. Alaythawy, "Iraqi License Plate Recognition Based on Machine Learning," *Iraqi J. Inf. Commun. Technol.*, vol. 3, no. 4, pp. 1–10, 2020, doi: 10.31987/ijict.3.4.94.

[25] X. Zhang, X. Ni, Y. Deng, C. Jiang, and M. Maleki, "Chinese License Plate Recognition Using Machine and Deep Learning Models," in *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, IEEE, Jul. 2021, pp. 342–346. doi: 10.1109/PRML52754.2021.9520386.

[26] A. Imran, H. Z. U. Rehman, M. Zakwan, and Z. Mehmood, "Performance Enhancement Method for Multiple License Plate Recognition in Challenging Environment," 2021.

[27] I. Mihoub, "Algerian license plate detection and recognition using deep learning," 2023.

[28] C. Yang, J. Jiang, J. Zhou, M. Hitosug, and Z. Wang, "Traffic safety and public health in China–Past knowledge, current status, and future directions," *Accid. Anal. Prev.*, vol. 192, p. 107272, 2023.

[29] Z. Zhu and Y. Wang, "A hybrid algorithm for automatic segmentation of slowly moving objects," *AEU-International J. Electron. Commun.*, vol. 66, no. 3, pp. 249–254, 2012.

[30] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proceedings fourth IEEE workshop on applications of computer vision. WACV'98 (Cat. No. 98EX201)*, IEEE, 1998, pp. 8–14.

[31] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, pp. 43–77, 1994.

[32] B. Aïssa, "Models for images and video foreground segmentation using finite mixtures of generalized Gaussians." ESI, 2016.

[33] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, p. 100204, 2020.

[34] J. Zhou and O. Veksler, "Color Separation for Background Subtraction," no. December, 2016.

[35] S. Singh Rawat, A. Sharma, and R. Gusain, "Analysis of Image Preprocessing Techniques To Improve Ocr of Garhwali Text Obtained Using the Hindi Tesseract Model," *Ictact J. Image Video Process.*, vol. 12, no. 02, pp. 2588–2594, 2021, doi: 10.21917/ijivp.2021.0366.

[36] K. A. M. Said and A. B. Jambek, "Analysis of image processing using morphological erosion and dilation," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 12033.

[37] R. Srisha and A. Khan, "Morphological Operations for Image Processing : Understanding and its Applications," *NCVSComs-13*, no. December, pp. 17–19, 2013.

[38] S. INTHIYAZ, S. K. H. AHAMMAD, A. KRISHNA, V. Bhargavi, D. Govardhan, and V. Rajesh, "YOLO (YOU ONLY LOOK ONCE) Making Object detection work in Medical Imaging on Convolution detection System.," *Int. J. Pharm. Res.*, vol. 12, no. 2, 2020.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[40] J. Of, M. Of, and M. Sciences, "IRAQI LICENSE PLATE RECOGNITION SYSTEM USING (YOLO) WITH SIFT AND SURF ALGORITHM University of Baghdad , College of Science , Department of Computer Science , Baghdad , Iraq University of Baghdad , College of Science , Department of Remote Sensing and G," pp. 545–561, 2020.

[41] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time license plate detection for non-helmeted motorcyclist using YOLO," *Ict Express*, vol. 7, no. 1, pp. 104–109, 2021.

[42] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 658–666, 2019, doi: 10.1109/CVPR.2019.00075.

[43] G. Tong, Y. Li, H. Gao, H. Chen, H. Wang, and X. Yang, "MA-CRNN: a multi-scale attention CRNN for Chinese text line recognition in natural scenes," *Int. J. Doc. Anal. Recognit.*, vol. 23, no. 2, pp. 103–114, 2020, doi: 10.1007/s10032-019-00348-7.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern*

Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[46] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv Prepr. arXiv1409.0473*, 2014.

[47] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[48] T. A. I. Abdelaziz and U. Fazil, "Applications of integration of AI-based Optical Character Recognition (OCR) and Generative AI in Document Understanding and Processing," *Appl. Res. Artif. Intell. Cloud Comput.*, vol. 6, no. 11, pp. 1–16, 2023.

[49] S. Patil *et al.*, "Enhancing optical character recognition on images with mixed text using semantic segmentation," *J. Sens. Actuator Networks*, vol. 11, no. 4, p. 63, 2022.

[50] H. R. Farhan, M. H. Al-Muifraje, and T. R. Saeed, "A new model for pattern recognition," *Comput. Electr. Eng.*, vol. 83, p. 106602, May 2020, doi: 10.1016/j.compeleceng.2020.106602.

[51] A. Kuznetsova *et al.*, "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, 2020, doi: 10.1007/s11263-020-01316-z.

Accepted papers

1. “An Efficient System for Detecting Multiple Traffic Violations and Recognizing License Plates using Video Processing and Deep Learning” in International Journal of Electrical and Electronic Engineering & Telecommunications (IJEETC).
2. “Vehicle License Plates Detection and Recognition” in the 7th international conference on engineering sciences (ICES) AIP Proceeding.

الخلاصة

تسبب المخالفات المرورية مشاكل كبيرة مثل الازدحام والحوادث والوفيات. ومن المرغوب فيه للغاية أن يكون هناك نظام آلي فعال لكشف وتسجيل هذه الانتهاكات، وبالتالي تحسين إنفاذ تنظيم المرور والحد من التدخل البشري. تهدف هذه الدراسة إلى كشف المخالفات المرورية وتحديد لوحات المركبات المخالفة في العراق. حيث يقوم النظام برصد المخالفات المرورية بشكل تلقائي، من ثلاث حالات مختلفة؛ كعدم الالتزام بإشارة المرور، وتجاوز الحد الأقصى للسرعة، والقيادة في الاتجاه المعاكس، فهي فعالة من حيث التكلفة وعملية وقوية. يستخدم النظام المقترح تقنية الطرح الخلفي لكشف المركبات المتحركة ومفهوم الفارق الزمني والمسافة التي تتحرك عبرها المركبات لكشف المخالفات. يتم استخدام خوارزمية "أنت تنظر مرة واحدة فقط" الإصدار 4 (YOLOv4) لتحديد لوحات الترخيص (LPs) للمركبات المخالفة بدقة كبيرة. تم استخدام الشبكات العصبية التلافيفية المتكررة (CRNN) مع تقنية التعرف الضوئي على الحروف (OCR) للتعرف على نوعين من الحروف المحدودة العراقية؛ النوع الأول يحتوي على أرقام هندية، أما النموذج الثاني فيحتوي على أرقام عربية، وهو نظام مستقبلي سيتم استخدامه في جميع أنحاء العراق. تشير النتائج التي تم تحقيقها من تجربتنا إلى أداء واعد للنظام، مع اكتشاف العديد من الانتهاكات في الوقت الفعلي ومعدل دقة إجمالي قدره 98.06% للمخالفات. ومن ناحية أخرى، فقد تفوق النظام المقترح في التعرف على LPs، حيث حقق نسبة نجاح بلغت 94.87% و98.22% للنوعين الأول والثاني على التوالي. وقد تفوقت على الأنظمة المماثلة، لا سيما من حيث الدقة والقدرة على اكتشاف أنواع متعددة من المخالفات المرورية في وقت واحد، مما أدى إلى ترسيخ قدرتها التنافسية.

جمهورية العراق

وزارة التعليم العالي و البحث العلمي

جامعة كربلاء

كلية الهندسة

قسم الهندسة الكهربائية والإلكترونية

نظام الكشف التلقائي عن مخالفات قواعد المرور

رسالة مقدمة الى مجلس كلية الهندسة / جامعة كربلاء وهي جزء من متطلبات نيل درجة الماجستير في علوم الهندسة الكهربائية

من قبل:

علي قاسم عبد علي

باشراف :

أ.م.د حميد رسول فرحان

أ.م.د مؤيد سليم كود

ذو الحجة - 1445 حزيران - 2024