drones

Article

FRCNN-Based Reinforcement Learning for Real-Time Vehicle Detection, Tracking and Geolocation from UAS

Chandra Has Singh [1], Vishal Mishra [1], Kamal Jain [1,2] and Anoop Kumar Shukla [3,*]

[1] Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, Uttarakhand, India
[2] Centre of Excellence Disaster Mitigation and Management, Indian Institute of Technology Roorkee, Roorkee 247667, Uttarakhand, India
[3] Manipal School of Architecture and Planning, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India
* Correspondence: anoop.shukla@manipal.edu

Abstract: In the last few years, uncrewed aerial systems (UASs) have been broadly employed for many applications including urban traffic monitoring. However, in the detection, tracking, and geolocation of moving vehicles using UAVs there are problems to be encountered such as low-accuracy sensors, complex scenes, small object sizes, and motion-induced noises. To address these problems, this study presents an intelligent, self-optimised, real-time framework for automated vehicle detection, tracking, and geolocation in UAV-acquired images which enlist detection, location, and tracking features to improve the final decision. The noise is initially reduced by applying the proposed adaptive filtering, which makes the detection algorithm more versatile. Thereafter, in the detection step, top-hat and bottom-hat transformations are used, assisted by the Overlapped Segmentation-Based Morphological Operation (OSBMO). Following the detection phase, the background regions are obliterated through an analysis of the motion feature points of the obtained object regions using a method that is a conjugation between the Kanade–Lucas–Tomasi (KLT) trackers and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering. The procured object features are clustered into separate objects on the basis of their motion characteristics. Finally, the vehicle labels are designated to their corresponding cluster trajectories by employing an efficient reinforcement connecting algorithm. The policy-making possibilities of the reinforcement connecting algorithm are evaluated. The Fast Regional Convolutional Neural Network (Fast-RCNN) is designed and trained on a small collection of samples, then utilised for removing the wrong targets. The proposed framework was tested on videos acquired through various scenarios. The methodology illustrates its capacity through the automatic supervision of target vehicles in real-world trials, which demonstrates its potential applications in intelligent transport systems and other surveillance applications.

Keywords: UAV; vehicle detection; tracking; geolocation; overlapped segmentation-based morphological operation (OSBMO); DBSCAN clustering; reinforcement learning (RL); Fast Regional Convolutional Neural Network (F-RCNN)

Citation: Singh, C.H.; Mishra, V.; Jain, K.; Shukla, A.K. FRCNN-Based Reinforcement Learning for Real-Time Vehicle Detection, Tracking and Geolocation from UAS. *Drones* **2022**, *6*, 406. https://doi.org/10.3390/drones6120406

Academic Editors: Marija Popović and Inkyu Sa

Received: 31 October 2022
Accepted: 29 November 2022
Published: 9 December 2022

1. Introduction

The success of automated airborne vehicles (UAVs) coupled with picture-handling calculations has prompted the extension of the application fields of UAVs. Utilising UAVs to recognise, track, or geolocate moving vehicles has drawn attention to a legitimate concern for scientists. These types of robots with following and geolocating structures have accomplished vast achievements in a rush-hour gridlock well-being assessment, a street surface check, traffic stream observations, and metropolitan security assurance [1,2] due to these robots being unaffected by ground gridlock. Additionally, they can deftly respond to scene changes as soon as it is possible to do so [3,4].

The ability to locate and follow vehicles is significant for security and reconnaissance applications as well as for Intelligent Transportation Systems. Recently, there has been an

expanded utilisation of automated airborne vehicles (UAVs) or drones for a reconnaissance due to their ability to observe far-off scenes [5,6]. However, with an increasing number of applications many challenges have appeared. The resolution of objects situated at long distances from the camera is low. Additionally, obscured parts and noise can deteriorate the picture quality. Different examinations have been focused on visual recognition and the following of moving articles and have used techniques such as background deduction and edge distinction. Recently, researchers have tried to address these concerns using many computer vision and deep learning methods. The Gaussian Mixture Model (GMM) was utilised to dissect the foundation and target districts in [7,8]. A Scale-Invariant Feature Transformation (Filter) was used to extricate closer-view objects from the background scene in [9]. The background was removed under Gaussian Mixture supposition, which was followed by the use of morphological channels [3,10]. Speculation fitting was embraced for vehicle recognition. Long-range moving items were distinguished using background deduction [4,11].

Notwithstanding the previously mentioned research, many investigations have been directed at following numerous objects [12,13]. Attempting to easily find numerous fast-moving objects causes a weighty mess (phony problem), and there is a low likelihood of discovery. The switching Kalman filter provides a solution for continuously assessing the condition of an objective [14,15]. Knowing—rather than assuming—the free Gaussian noise element is ideal. At the point where different estimations are distinguished at the edge, the information affiliation is expected to dole out the estimations to the laid-out tracks [6].

Despite different investigations having been conducted on acquiring vehicle locations from UAVs, some problems still remain. For example:

1. The object density is high in complex urban scenarios such as densely filled parking areas, at intersections, or in clogged streets, and determining the location of an individual vehicle can become troublesome [16]. Additionally, vehicles might be blocked by trees, boards, or different developments to some extent. Different variables that plague the recognition of vehicles include complex backgrounds, shadows, shifting light conditions, and distinctions in the vehicles' types, appearances, and directions. This multitude of variables lessens the adequacy of the usual methods such as an optical stream, outline distinction, and foundation deduction.
2. Differences in the top aerial view and the terrestrial view of the vehicle make detection more challenging. The airborne pictures lack the front-view physiognomy of the vehicle and vehicles show rectilinear shapes in the top view. Another change observed in the aerial imagery is that of scale (resolution). The size of vehicles when captured from UAVs is small compared to normal ground images. For instance, in a 5K $\times$ 3K pixel image captured from a UAS, a vehicle might appear at 50 $\times$ 50 pixels. Therefore, it becomes challenging to detect the vehicle as it is difficult to find the variations in its features that distinguish it from other similar-looking vehicles. Resolution also makes it challenging to differentiate vehicles from other objects such as big containers, garbage bins, street signs, and other rectilinear objects.
3. Low-elevation UAVs are more plagued by sudden movements and natural variables. Given that the camera perspectives and flying elevations of more modest UAVs change quickly, the information they acquire fluctuates significantly. Moreover, the carrying capacity of small UAVs restricts the weight of the computational hardware they can fly with onboard.

To overcome the above challenges for the precise execution of detecting, tracking, and geolocating a vehicle, an intelligent, self-improved, constant methodology was developed for automated vehicle identification, following, and geolocation in UAS-acquired images that utilise recognitions, area, and following elements to upgrade an ultimate choice. This work aims to present an intelligent, self-optimised, real-time approach for automated vehicle detection, tracking, and geolocation in UAV images that utilise detections, location, and tracking features to strengthen the final decision. The main contributions of this research are:

(1) A proposed adaptive filtering method for reducing noise which enhances the reliability of the detection algorithm;
(2) To develop a top–bottom-hat transformation assisted by the Overlapped Segmentation-Based Morphological Operation, which is to be employed in the detection phase;
(3) To initiate the elimination of background regions by motion–feature point analysis of the obtained object regions using a conjugated technique of DBSCAN clustering and KLT trackers;
(4) To develop an efficient reinforcement-connecting algorithm for assigning the vehicle labels corresponding to their cluster trajectories.

The rest of this paper has been composed as described in subsequent lines. Section 2 describes the previous studies; Section 3 presents the nuances of the proposed approach; Section 4 discusses the display appraisal; and Section 5 concludes the proposed work.

2. Previous Studies

Zhao et al. [17] developed a system for moving vehicle recognition, following, and geolocation using a monocular camera, a Global Positioning System (GPS) collector, and sensors for inertial measurement units (IMUs). Initially, the strategy utilised YOLOv3 [18] for vehicle recognition due to its adequacy and proficiency in discovering small objects in complex scenes. Subsequently, a visual tracking strategy considering connection channels was presented, and a latent geolocation technique was introduced to compute the GPS directions of the moving vehicle. Finally, a flight-control technique was introduced to lead the UAV that follows the vehicle of interest. This methodology was implemented on a DJI M100 stage to which a microcomputer Jetson TX1 and a monocular camera were added. The exploratory outcomes showed that the UAV was equipped for identifying, following, and geolocating the vehicle of interest with high accuracy. The structure exhibited its ability in programmed oversight on tracked vehicles with genuine analyses, which recommended its possible applications in urban rush-hour gridlock, planned operations, and security.

Avola et al. [13] presented an efficient, novel multi-stream (MS) algorithm. The algorithm included the application of different kernel sizes to each stream for performing image analysis on multiple scales. The proposed design was then utilised as the spine for the notable Faster R-CNN processing, characterising a MS–Faster R-CNN object locator that reliably identifies objects in video groupings. This locator was mutually utilised with the Basic On the Web and Continuous Following a Profound Affiliation Metric (Profound SORT) calculation to accomplish constant following capacities for UAV pictures. Extensive tests were performed on the different UAV datasets to assess the proposed methodology. The introduced pipeline has achieved best-in-class performance, confirming that the proposed multi-stream strategy is robust and can accurately mimic the multiscale picture examination worldview.

A technique based on the Kanade–Lucas optical flow method was proposed by Valappil et al. [19] for detecting a moving object. This was followed by isolating the objects by building connected graphs. This was then followed by a convolutional neural network (CNN), trailed by a Support Vector Machine (SVM) for definite grouping. The optical stream created contains foundation (and small) objects identified as vehicles as the camera stage moves. The classifier presented here prevents the presence of some other (moving) objects from being identified as vehicles. The method being described was tested on stationary videos and moving aerial recordings.

Li et al. [20] fostered a methodology that included: (1) a deep deterministic policy gradient (DDPG)-based control system to provide learning and independent dynamic capacity for UAVs; (2) a superior technique named MN-DDPG that presented a kind of blended noise to help UAVs with stochastic policies for ideal online preparation; and (3) a calculation of errand decay and pre-preparing for proficient exchange, determining how to further develop the speculation capacity of a UAV's control model constructed in view of MN-DDPG. The results of the trial recreation verified that the methodology yielded a

significant improvement in the ability to anticipate critical self-manoeuvrability changes in the UAV's flight behaviour and the effectiveness of the UAV designed to operate in regulatory ventures in suspicious environments.

A hybrid vision-based framework was proposed by Espsoito et al. [21] to independently recognise and follow a UAV with a moving camera. A Faster Region-based Convolutional Neural Network (Faster R-CNN) was designed and exploited in the detection stage to distinguish the Region of Interest (RoI). The UAV's location in the image plane was determined by this RoI. The moving object was followed by an optical flow-based tracking framework and a Kalman filter was utilised to give fleeting consistency between back-to-back estimations. The global positioning framework was intended to have the option to accomplish constant picture handling on implanted frameworks; therefore a slack pay calculation for the deferral due to the Faster R-CNN calculation time was carried out. The performance of the proposed model was assessed by the deviation between the genuine UAV position in the image plane and the assessed position estimated from the positioning framework.

Shao et al. [22] developed a group-movement assessment framework due to the variable-scale corner detection and optical stream that utilised the infrared cameras and high adaptability of the UAV. An airborne infrared imager, TAU2-336, was used to capture the original images. Median filtering was used for pre-processing the infrared images. Afterwards, multiscale analysis was employed for corner detection and tracking. The average velocity of the crowd was estimated in the final stage. The trial results showed that the methodology was viable for assessing the crowd movement speed and behaviour.

However, in [13,17,19], the research does not focus on the detection, tracking, and geolocation of moving vehicles based on the airborne platform; it has been suffering from small object sizes and, in some of the studies [20,21], scene complexity was a major problem. In [22], the research falls short of precise detection and contains low-accuracy sensors. Given the consideration mentioned above, there is a great necessity to develop a novel strategy for the superior expansion of UAVs. Table 1 encapsulates the novelty and shortcomings of the existing studies.

Moreover, the conventional methods that were utilised in the existing literature are lacking in many aspects and therefore novel techniques are very much needed for further processing and improvement. Hence, this work employs reinforcement learning to overcome these issues. Reinforcement learning (RL) is a subset of the machine learning concept that deals with the multi-state decision-making of a software agent (in this case, a UAV) as it interacts with its surroundings. The proposed framework employs RL as it will maximise resource utilisation to carry out intelligent vehicle recognition and localisation more effectively. It blends the decision-making skill of reinforcement learning with the perceptual capacity of deep learning.

With the aim of developing a faster algorithm for real-time automated detection of vehicles from UAVs, we have developed a framework based on FRCNN [23]. To improve its performance we have introduced an adaptive filter, enhanced the detection by utilising top-hat and bottom-hat transformations assisted by the OSBMO, and finally employed a conjugated technique of DBSCAN clustering and KLT trackers to eliminate the background regions. After these steps, RL based on a Fast R CNN-OMDP was used for vehicle identification. We evaluated this framework on our collected UAV-viewed test dataset and classical dataset.

Table 1. Features and challenges of the existing studies.

Author [Citation]	Techniques	Features	Challenges
Avola et al. [13]	Faster R-CNN	• Enhanced precision	• Needs improvement in speed without penalising the model accuracy
Zhao et al. [17]	YOLOv3	• Cost effective • Larger performance	• Need for navigation algorithm to improve the system with more intelligence
Valappil et al. [19]	CNN-SVM	• Improved accuracy • Excellent performance for low-to medium-congested traffic conditions	• Multi-class classification using deep learning approaches needs improvement
Li et al. [20]	MN-DDPG	• Improved convergence speed • Enhanced the generalisation capability of the UAV control model	• Further enhancement needs in evaluating the efficiency, robustness, and performance
Espsoito et al. [21]	Faster R-CNN	• Significantly low error • Better accuracy	• Need to exploit more on tracking system information
Shao et al. [22]	Crowd motion estimation system	• Better evaluation of crowd behaviour status and motion speed	• Creates false noise edges

3. Proposed Methodology

The utilisation of UAV-acquired images (video frames) for real-time automatic detection and tracking of moving vehicles is a challenging issue due to vehicle impediments, camera development, and high computational expense. This work presents an intelligent, self-optimised, continuous methodology for programmed vehicle identification, following, and geolocation in elevated pictures that utilises recognitions, areas, and following highlights to upgrade an official choice, as is displayed in Figure 1.

3.1. Noise Reduction

Initially, the quality of the images was upgraded by pre-processing. The pre-processing further contributes to developing precise vehicle identification and following. Pre-processing is completed by utilising this strategy, for example.

The pre-processing of the image is given as:

$$\Im_{(i,j)} = \chi_{i,j} \begin{bmatrix} S_{(1,1)} & S_{(1,2)} & S_{(1,3)} \\ S_{(2,1)} & S_{(2,2)} & S_{(2,3)} \\ S_{(3,1)} & S_{(3,2)} & S_{(3,3)} \end{bmatrix} \quad (1)$$

where $\chi_{i,j}$ is the pre-processing function performed over an image $S_{(i,j)}$.

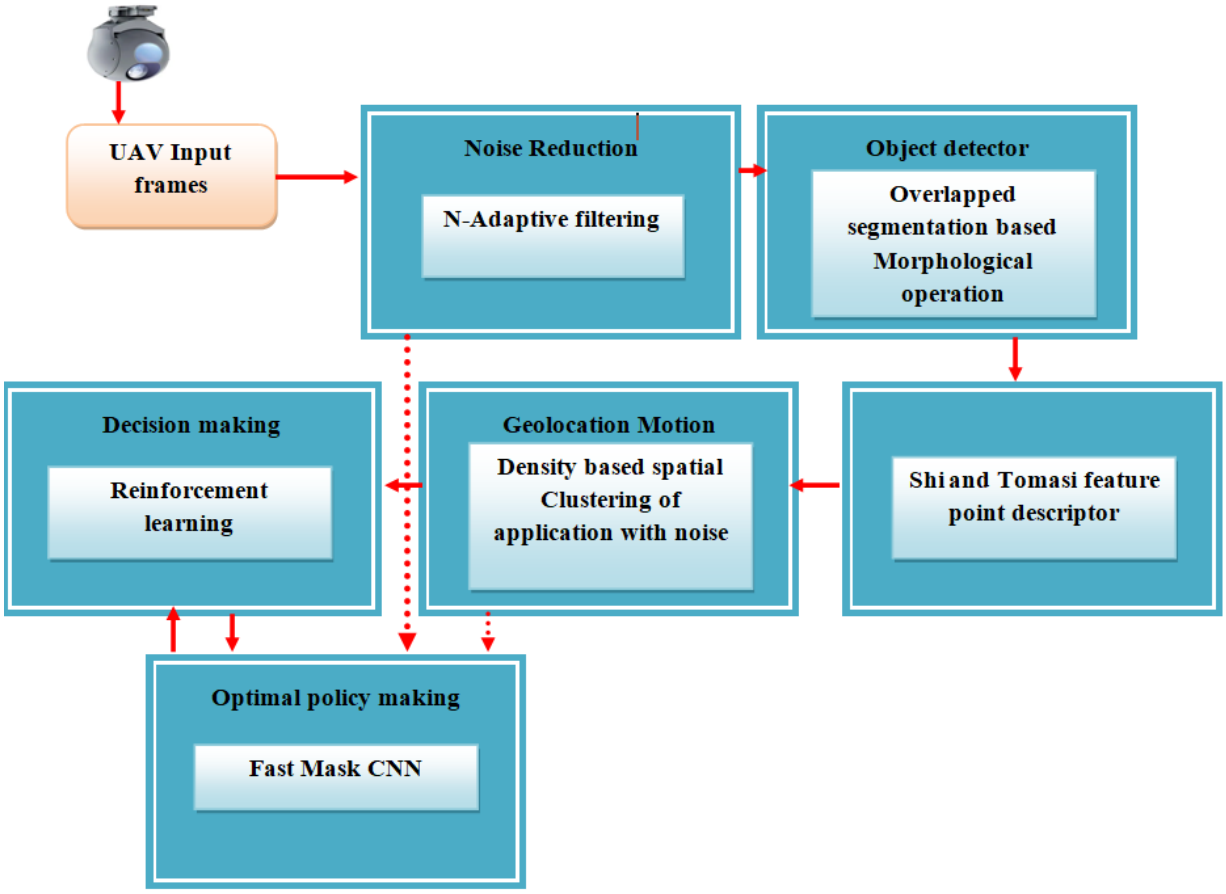


Figure 1. Proposed methodology.

(a) Denoising

Denoising, as is evident from the term, is the process is performed to remove noise from a picture to re-establish its primary information. The denoising of a loud image provides a productive method for recognising the essence of the image. The denoising of an image is provided by:

$$\chi_{1(i,j)} = \chi_{den(i,j)}\left[S_{i,j}\right] \tag{2}$$

where $\chi_{den(i,j)}$ is the number of denoising functions used in the proposed work.

This work utilises a Standard Deviation-based Middle Channel (SD-MF+BF) and respective separating to denoise the picture.

First, the image is handled under pre-processing to expel drive noise, etc. Motivation noise defiles the sharpness of the data and will, in general, corrupt the nature of the picture. Regardless of different noise removal methods that have been established, some imperfections always remain, for example, the loss of information, wavers when the likelihood of drive clamour is high, etc. This framework has employed a Standard Deviation-based Versatile Middle Channel (SD-MF+BF) to capture the current strategies.

Initially, the picture Γ is transformed into a grey-level picture. This grey-level data is represented by $\Gamma_{i,j}$ and the grey levels of a noisy image are represented by $E_{i,j}$. $[I_{min}, I_{max}]$ indicate the powerful scope of picture dim levels. In the first 8-cycle changes, σ^2(variance) and S (standard deviation) are assessed for every original pixel. The range of evaluation lies between $I_{min} = 0$ and $I_{max} = 255$ for variance observed, and it is standardised with a value of $N = 255$.

$$\sigma^2 = \frac{\sum_{i,j=0}^{n} \left(\Gamma_{i,j} - \mu\right)^2}{N} \tag{3}$$

$$S = \sqrt{\sigma} \tag{4}$$

On the basis of the standard deviation, an ordinary conveyance bend is estimated and, considering the bend, 95% of the information lies in between the first and second standard deviations and whose exact formulae $\mu + 2\sigma$ are observed and assessed by:

$$Y_{ND} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\Gamma-\mu)^2}{2\sigma^2}} \tag{5}$$

Following this, the formulation of noise is performed based on 95% of picture pixels with the likelihood p that are depicted in the following function:

$$\mathrm{E}_{ij} = \begin{cases} \mathrm{I}_{ij} & with\ probability\ p \\ \Gamma_{ij} & with\ probability\ 1-p \end{cases} \tag{6}$$

where I_{ij} signifies the ordinarily dispersed pixels inside the reach $[\mathrm{I}_{\min}, \mathrm{I}_{\max}]$, and can be any number between $\mathrm{I}_{\min}$ and $\mathrm{I}_{\max}$.

The median values are used by the filter to replace noisy pixels. The median value is the figure lying in the middle of the arranged grouping of values. The grey values of any pixel value in any window (W_e) of size $n \times n$ are characterised by $e_1, e_2, e_3, e_4, \ldots\ldots e_n$, which becomes $e_{i1} \leq e_{i2} \leq e_{i3} \leq e_{i4} \leq \ldots\ldots e_{in}$ subsequent to arranging the values in ascending or in descending phenomenon.

$$W_e = median(W_e) = \begin{cases} \mathrm{E}_{i(n+1)/2} & n\ is\ odd \\ \frac{1}{2}\left[\mathrm{E}_{i(\frac{n}{2})} + \mathrm{E}_{i(\frac{n}{2})} + 1\right] & n\ is\ even \end{cases} \tag{7}$$

Finally, the derived image is freed from drive noise with nearly no adulteration of values.

A reciprocal channel is a non-direct channel which has an edge-defending property, close-by clatter clearing. This channel is good for wiping out upheaval contents without over-darkening the image as well as protecting the image quality. The essential reason behind using an equal channel is that two pixels should be close to each other rather than expecting that they are accessible in nearby regions; moreover, their resemblance should be in photometric reach. The advantage of using an individual channel over an equivalent Gaussian channel is that the two-sided channel includes power assortments to defend its edges. It registers the weighted measure of pixels in a close-by region. For each adjacent pixel, a weighted ordinary is used for replacing the pixel regard. The consequence of a proportional channel for a pixel X can be framed using Equation (8) as:

$$\Gamma_W(W = P) = \frac{1}{W_P} \sum_{Q \in \alpha} G_\alpha(||P - Q||) F_R\left(L_P - L_Q\right) L_Q \tag{8}$$

This is a normalised, weighted typical where P and Q are the pixel co-ordinates, α addresses the spatial neighbourhood of $\Gamma_W(W)$, G_α is a spatial Gaussian that reduces the effect of distant pixels, and F_R is an approach at a Gaussian that decreases the effect of pixels Q when their power values differ from the power of pixel P, L_P. W_P is the normalisation factor which can be enlisted using Equation (9) as:

$$W_P = \sum_{Q \in \alpha} G_\alpha(||P - Q||) F_R\left(L_P - L_Q\right) \tag{9}$$

Consider a pixel with coordinates (i, j) which is to be denoised utilising the two-sided channel and let (k, l) be the adjoining pixel co-ordinate. The load to be appointed to pixel (k, l) to denoise the pixel at (i, j) is determined utilising Equation (10) as:

$$\Phi(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\beta_d^2} - \frac{||L(i,j) - L(k,l)||^2}{2\beta_R^2}\right) \tag{10}$$

Here, β_R and β_d are the smoothing boundaries, and $L(i,j)$ and $L(k,l)$ are the pixel powers. β_R and β_d are the two boundaries that control the behaviour of the respective channel. β_R determines the power–area conduct of the two-sided channel and β_d indicates the spatial conduct of the reciprocal channel.

3.2. Object Detector

The denoising may compel an overlapping of vehicles. If the covering vehicle is not considered, then there may be a probability of a high error rate while identifying the vehicle. A measurable examination has been performed in view of the denoised picture utilising a *t*-test to assess the covering object.

First, the covering clusters $\left(O^{\Phi}_{ful}, O^{\Phi}_{par}, O^{\Phi}_{nor}, O^{\Phi}_{1/4},\right)$ are viewed as edges that demand completely, to some degree, ordinary and $\frac{1}{4}$-object covering over some haphazard outlines. $\mathrm{T}_{sample} = \frac{O^{\Phi}_{n}}{\mathrm{B}}$ is chosen in light of the group size of (B). In view of the populace mean $\left(\overline{O}^{\Phi}_{n}\right)$ and test mean $\left(\overline{\mathrm{T}}_{sample}\right)$, the *t*-test is used. The distinction between the two clusters is estimated by *t*-test (t) figures. Following null hypothesis H0, an alternative hypothesis is formulated prior to figuring out the *t*-test:

H0 : *There is a vast difference between the population mean and sample mean; that is, the cells are overlapped.*

H1 : *There is no difference between the population mean and sample mean; that is, the cells are not overlapped.*

$$t = \frac{\overline{\mathrm{T}}_{sample} - \overline{O}^{\Phi}_{n}}{SD/\sqrt{\mathrm{B}}} \tag{11}$$

There is a 'significant value', also called a *p*-value, in the obtained *t*-test, which is a likelihood that comes from the sample information occurred by some coincidence. The overlapping of cells is estimated on the basis of the *p*-value. The magnitude of the *p*-value determines the affirmation of the null hypothesis. If it exceeds the value of 0.05, the alternate hypothesis is rejected and the null hypothesis is approved. Therefore, overlying cells can be determined due to the *t*-test.

From the results of the *t*-test, the covered cell from the particular populace is isolated to shape a solitary cell utilising dilation and erosion operations. At this point, a high blunder rate may occur in the unlikely instance that the covered objects are not isolated. The covered objects are isolated utilising disintegration and widening.

(a) Erosion:

The covered picture goes through an erosion O with an organising component (meant $O\Theta\wp$), making another picture $G = O\Theta\wp$ in all places (x,y) in which that organising component $\wp$ matches the information picture O. For example, $G(x,y) = 1$ if O fits $\wp$, is generally zero in any remaining spots, and replays for all the pixels.

Erosion takes out limited-scope data from a paired picture while lessening the area of concern at the same time. The limits of every region might be found by eliminating the dissolved picture from the first picture:

$$B = \wp - (\wp\Theta O) \tag{12}$$

where O is an image of the regions, O is a 3-3 structure variable, and B is an image of regional boundaries.

(b) Dilation:

The clustered picture goes through a dilation with an organising component making another parallel picture $G = \wp \oplus O$ in all spots (x,y) in which that organising component C matches the info picture $\wp$. For example, $G(x,y) = 1$ if O fits $\wp$, is otherwise zero in any remaining spots, and goes on for all pixel–arrays. The erosion and dilation produce a solitary

cell. Furthermore, the cells are tested using the *t*-test to determine if there are any covered cells present and, if there are not, the extraction of highlights is finished for the picture.

3.3. Feature Point Descriptor

The Kanade–Lucas–Tomasi (KLT) highlight tracker is a methodology utilised to identify movement by feature selection and extraction. A component, or a focal point, is a point or collection of places where the algorithm can search and track the movement through outlines. For tracking features across image frames it is of utmost importance for the purpose of error minimisation that "good" features are selected. The matrix for choosing Shi–Tomasi features is as follows:

$$Z = \sum_{PY-WY}^{PS+WS} \sum_{PX-WX}^{PX+WX} \begin{bmatrix} I_Y^2 & I_Y I_X \\ I_Y I_X & I_X^2 \end{bmatrix} \tag{13}$$

where Z is the second framework of the picture I about a point $U = (PX, PY)$ with the window W of size $(2WX + 1) \times (2WY + 1)$. The points U_i in the picture I, where G is non-solitary and the base eigenvalue $\hbar_{\min} = \min(\hbar 1, \hbar 2)$ of G is over a particular edge, $\hbar^{th}$, are considered to be the interest points. In the wake of recognising the interest points in outlines $I(X, Y, t)$ and $I(X, Y, t + 1)$, an interest point U_i can be followed from time t to time $t + 1$ with the Kanade–Lucas calculation for optical stream. For following points across distances on the request for a few pixels, an iterative execution with picture pyramids was utilised. Consider I_L the pyramidal picture of I at the pyramid level L.

$$U^L = \frac{U^*}{2^L} \tag{14}$$

where U_i is any point in I. The ideal dislodging S^* can then be assessed by limiting the blunder capability $\in (SX, SY)$:

$$S^* = \text{argmin}[\in (S_X, S_Y)] \tag{15}$$

The mistake capability is:

$$\in (S_X, S_Y) = \sum_{PY-WY}^{PS+WS} \sum_{PX-WX}^{PX+WX} \left[I^L(P_X, P_Y) - J^L(P_X + S_X, P_Y + S_Y)^2 \right] \tag{16}$$

3.4. Geolocation Motion Detector

The feature point descriptor from the discovery strategy is trailed by geolocation movement detection. Movement identification assists with finding the vehicle district from a unique region. This work has fostered an introduced thickness-based spatial grouping of utilisation with commotion. The technique is equipped for following movement in the picture as well as video outlines. It gives a grouping of the comparable area and recognises the geolocation and, from that point, plays out the bounding box creation over the movement region.

Initially, generation of the histogram information of the picture is performed and, following discovery, is transferred to a semi-administered multi-object that distinguishes the exceptional similitudes between the pixel points in an image and groups them per likeness.

Two parameters that dictate this methodology are minimum points (MinPts) and Epsilon. The radius of a circle is stated by Epsilon (ε) by considering a one-pixel point from the image. The number of points at which the condition of formation of clusters is satisfied is defined by MinPts.

Given the MinPts and Eps, three significant points are defined; namely, noise points, boundary points, and core points. A pixel point is intended to be the centre point on the chance that it fulfils the MinPts inside the Eps Distance. A pixel point is designated as a boundary point on the chance that it is a neighbour of the centre point. If a point does not belong to the category of core point or boundary point then it is categorised as a noise point.

Euclidean distance forms the basis of the computation of the core point ($\forall^{\varepsilon}$), boundary point (B^{ε}), and noise point. Initially, a random pixel point is selected ($F\left(\mathrm{P}_{i,j}\right)$). The pixel point is then checked by drawing a circle of distance ε with a condition of satisfying the MinPts using Euclidean distance that is computed by:

$$\forall^{\varepsilon}\left[F\left(\mathrm{P}_{i,j}\right)\right] = \sqrt{\left(\mathrm{P}_{l}^{1} - \mathrm{P}_{m}^{1}\right)^{2} + \left(\mathrm{P}_{l}^{2} - \mathrm{P}_{m}^{2}\right)^{2} + \ldots \left(\mathrm{P}_{l}^{n} - \mathrm{P}_{m}^{n}\right)^{2}} \tag{17}$$

$$B^{\varepsilon}\left[F\left(\mathrm{P}_{i,j}\right)\right] = \sqrt{\left(B_{l}^{1} - \forall_{m}^{1}\right)^{2} + \left(B_{l}^{2} - \forall_{m}^{2}\right)^{2} + \ldots \left(B_{l}^{n} - \forall_{m}^{n}\right)^{2}} \tag{18}$$

The authenticated image clusters $\left(C_{K}^{I}\right)$ will be formed based on the boundary points and core points, and the change of motion is stated as an outlier $(C*)$.

Based on the distance between the truth and the predicted bounding box, the distance loss value is evaluated and the motion is detected.

3.5. Decision Making

Navigation outlines the marking of the item found through distinguishing the movement and the article. To furnish this strategy with ideal dynamics, a Fast R CNN-OMDP (Markov Decision Process with Option [24]) strategy for identifying the vehicles precisely has been developed. The portrayal of the proposed dynamic method is shown in Figure 2.

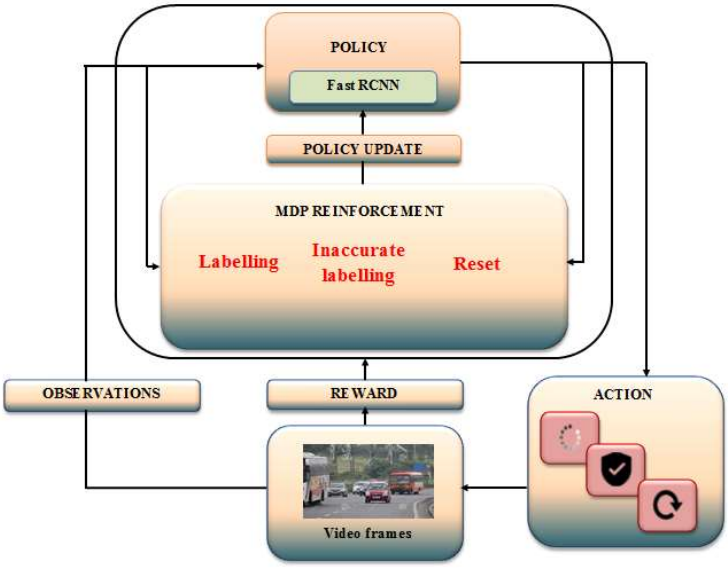


Figure 2. Proposed optimal reinforcement learning.

3.6. GKMF (Gaussian Kernel Membership Function) -OMDP Approach for Interval-Valued Decision Dystem (IDS)

Initially, the environments are categorised into three states: "Normally Operating (S_{N}^{1})", "Crowded (S_{A}^{2})", and "Overcrowded (S_{F}^{3})". Consider the respective actions performed over each state labelled as: "labelling (A_{W}^{1})", "unlabelled (A_{D}^{2})", and "reset (A_{R}^{3})", as are illustrated in Figure 3. Let P be the probability of transition for state–action pair (P_{SA}), i.e., $\mathrm{P}_{SA} \in \Gamma^{||S||}$, and let a reward function $\Re(S, \mathrm{A})$ be allotted for the correct state–action pair and respective discount factor $\Phi \in (0, 1)$. A policy π maps for each period $t \in N$ is assigned, and a state–action history up to time $t(S_0, \mathrm{A}_0, S_1, \mathrm{A}_1, S_2, \mathrm{A}_2, \ldots S_t \mathrm{A}_t)$ to a probability distribution over the set of actions ($\aleph_{S,\mathrm{A}}$) is initiated. In general, the policy

is history-dependent. The goal is to find a policy π that maximises the infinite horizon discounted expected reward $\Re(\pi, P)$:

$$\Re(\pi, P) = \mathrm{E}^{\pi,P}\left[\sum_{t=0}^{\infty} \Phi^t \pi_{S_t A_t} | S_0 = p_0\right] \tag{19}$$

where S_t represents the state at time period t and A_t illustrates the action chosen at time t that follows the probability distribution of $(\pi_{S_t \mathrm{A}_t})_A \in \Gamma^{||\mathrm{A}||}$. The vector $p_0 \in \Gamma^{||\mathrm{A}||}$ is a given initial probability distribution over the set of states S. it is assumed that $\mathrm{A}_s = \mathrm{A}$ for all states s and that the rewards are non-negative. It is also assumed that the set of states and the set of actions are finite. The OMDP architecture has been depicted in Figure 3.

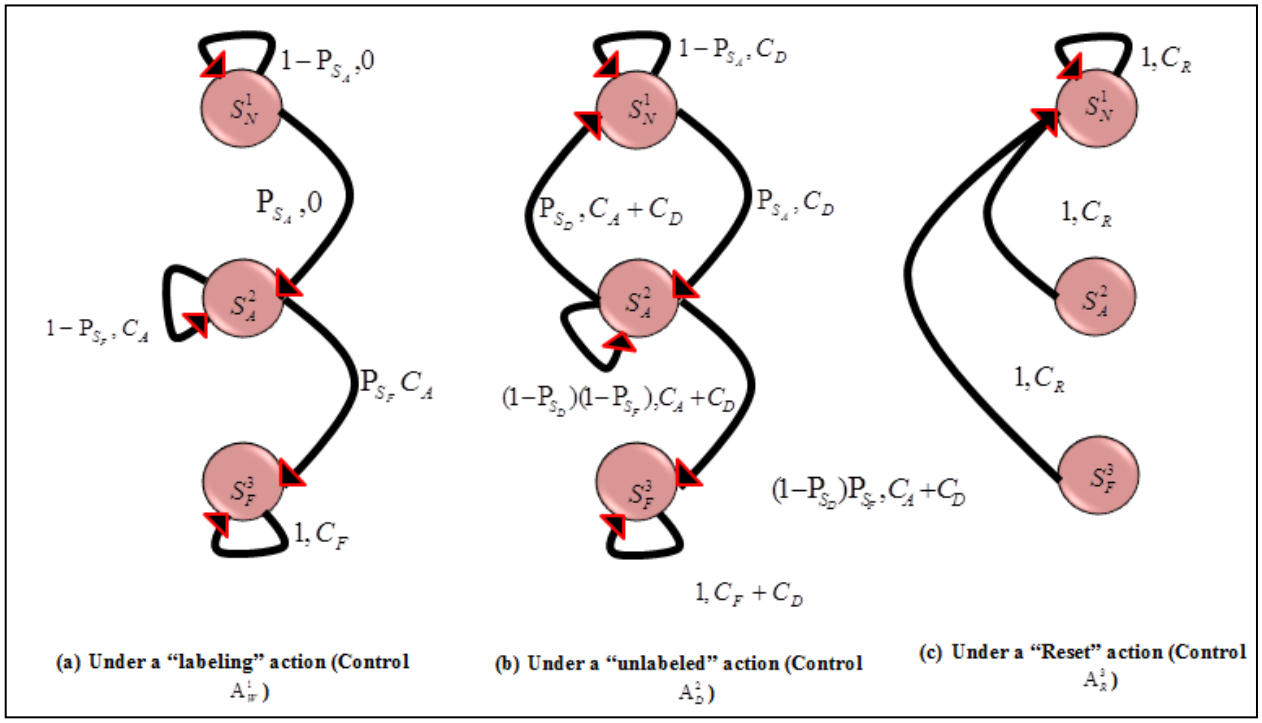


Figure 3. OMDP Architecture.

The object-detection system is built based on the transition among the states and the respective action outputs that will explicate whether the environment is normal or abnormal. Initially, under the "labelling" action (control A_W^1), there are self-transitions in the state S_N^1 that represent a secure environment and which illustrate the normal execution of the process. However, when there is a transition from state S_N^1 to S_A^2 occurring at per-stage probability P_{S_N}, then an intrusion attempt begins. Thereafter, self-transitions in the state S_A^2 are unlabelled. Eventually, a transition from state S_A^2 to S_F^3 occurs with per-stage probability P_{S_F}, which represents the starting of the reset that persists indefinitely. The cost of the transition beginning at the state S_N^1 is cost-free, whereas for the transition beginning at the state S_A^2 and S_F^3 there exists costs of C_A and C_F. These same probabilities or cost parameters apply under the "unlabelled" action (control A_D^2) but with two differences: first, the possible transition from the state S_A^2 back to S_N^1 (occurring with per-stage probability P_{A_F}) represents the successful disruption of the intrusion attempt via reducing the intrusion cost; and second, the cost of disruption C_D is incurred in addition to the transition cost under control A_W^1. Finally, under the "reset" action (control A_R^3), a transition back to state S_N^1 occurs with a probability of one, incurring at the beginning of the decision stage a cost of C_R no matter the state.

3.6.1. Optimal Policy for Decision Making

Based on the OMDP choice, an ideal strategy is implemented, which eventually positions the qualities from the picture ($\lambda_{Ranking}$) and video stage ($Z_{Ranking}$) to be given to the extremity characterisation, a move toward in training the dataset to rely on the extremity of the position value of the information. The current extremity characterisation will, in general, accomplish an erroneous recognition because of the low quality of the picture and the off-base division of the picture. To overcome these difficulties, a Mask R-CNN assembly to analyse the movement based on the geolocation has been developed (Figure 4). The created approach is split into specific layers, such as:

(1) Convolution layer
(2) RPN Layer
(3) ROI layer
(4) Segmentation mask layer
(5) Fully connected layer

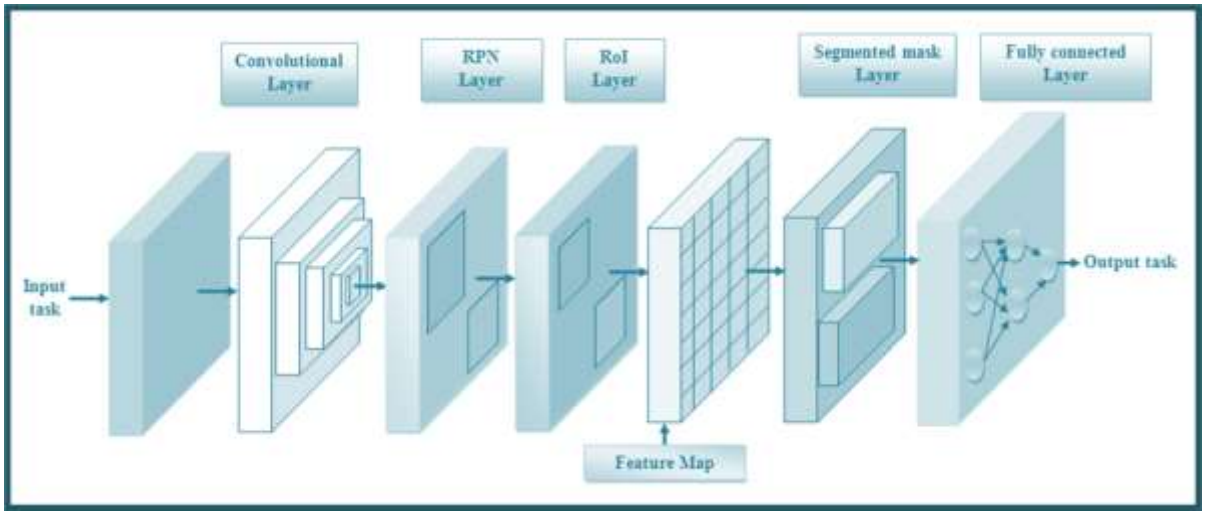


Figure 4. Fast RCNN.

3.6.2. Convolution Layer

Convolutional neural networks are a profound learning technique that take input text, grids, and pictures and convolve them with channels or pieces to extricate highlights. The information text network and the picture are convolved with a channel, and this convolution activity learns a similar component of the whole picture. The size of the resulting network with no padding is represented by:

$$\left[\lambda_{i,j}, Z_{i,j}\right] \times \xi_{i,j} = \lambda - \xi + 1 \tag{20}$$

The window slides after every activity and the highlights are advanced by the component maps. The element maps catch the neighbourhood's open field of the picture and work with shared loads and predispositions. The convolution activity is given by:

$$O_{CONV} = \sigma_{sigmoid}\left(b + \sum_{i=0}^{2}\sum_{j=0}^{2} W_{i,j} act_{a+i,b+j}\right) \tag{21}$$

Padding is utilised to safeguard the size of the information picture. In 'SAME' padding, the resultant picture size is equivalent to the information picture size and 'VALID' cushioning is regarded as no cushioning. The size of the resulting network with padding is stated as:

$$\left[\lambda_{i,j}, Z_{i,j}\right] \times \xi_{i,j} = (\lambda + 2p - \xi)/(\phi_s + 1) \tag{22}$$

Here, O_{CONV} is the output, p is the padding, ϕ_s is the stride, b is the bias, $\sigma_{sigmoid}$ is the sigmoid activation function, $W_{i,j}$ represents the weight matrix of shared weights and $act_{a+i,b+j}$ is the input activation at the position i, j.

After the padding of the resulting array, the convolution layer receives an element map for the concerning picture. The acquired component map is given by:

$$\bar{\Omega}_{FM} = \begin{bmatrix} \lambda_{i,j} \\ Z_{i,j} \end{bmatrix}_N, N \; is \; the \; no \; of \; feature \; maps \; of \; both \; text \; and \; image \tag{23}$$

3.6.3. RPN Layer

A Region Proposal Network (RPN) takes the element map as input and predicts whether the names and movements are present or not. The RPN utilises a sliding window to filter the component guides and track down the return for value-invested regions (cells) where the object exists. Each obtained return for value invested region is a square shape (anchor) on the image.

After the RPN network handling and forecast, a progression of bounding boxes can be obtained and their situation and size are remedied. On the slight chance that different bounding boxes cross over one another, the Non-max Suppression (NMS) is applied to obtain the jumping box with a higher forefront score and pass it on to the following stage.

There are two result layers for each RPN, i.e., a text/non-text characterisation layer and a rectangular bounding box relapse layer. The loss function of the RPN can be indicated as follows:

$$LOSS = L(\lambda_1, \lambda_1^*, B_1, B_1^*) + L(Z_2, Z_2^*, B_2, B_2^*) \tag{24}$$

$$LOSS[(\lambda, \lambda^*, Z, Z^*, B, B_1^*)] = L_{CLS}(\lambda, \lambda^*) L_{CLS}(Z, Z^*) + \lambda L_{REG}(B_1, B_1^*) \tag{25}$$

where λ, Z and B are the predicted labels of text, images, and boxes for RPN, λ^*, Z^* and B^* are the ground truth values of labels of text, image and boxes, L_{CLS} and L_{REG} are, respectively, the losses of classifier and regressor, and λ is the learning rate.

$$\forall_{RPN} = \begin{bmatrix} B_1(\lambda_{i,j}) \\ B_2(Z_{i,j}) \end{bmatrix}_N \tag{26}$$

Thus, bounding boxes are obtained by evaluating the loss and preceding the next stage.

3.6.4. ROI Layer

At this stage, the picture from the RPN is of various shapes, so the pooling layer is acquired to reshape it into a similar size. It adjusts the separated highlights to the first district proposition network appropriately and assists with delivering better pixel division results. The ROI layer assists with working on the exactness of the model.

For every one of the anticipated districts, the Intersection over Union (IoU) with the ground truth boxes for both text and picture input is figured. The IoU is given by:

$$IoU = \frac{\lambda \cap \lambda^*}{\lambda \cup \lambda^*} \tag{27}$$

Thereafter, provided that the IoU is more prominent than or equivalent to 0.5, it will be considered a region of interest. In any case, that specific region is dismissed. This process is repeated for every one of the locales and a few districts for which the IoU is more prominent than 0.5 are selected.

3.6.5. Segmentation Mask Layer

A resultant mask layer is added based on the previous layer. This provides the segmentation mask for every region that contains an item. The segmentation mask layer acquires the expectation of cover for all articles in a picture.

Finally, the division cover object is levelled and given as a contribution to a completely associated layer.

$$\forall_{flattened} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, Z_1, Z_2, Z_3] \tag{28}$$

3.6.6. Fully Connected Layer

The contribution to the completely associated layer results from the past division veil layer, which is levelled and afterwards taken care of by the completely associated layer. The straightened vector prepares the completely associated layer, such as that of ANN. The preparation of the vector is finished utilising:

$$\forall_I^T = act\left(\sum_{i=1}^{n} \nabla_i \forall_{flattened} + \aleph_b\right) \tag{29}$$

where $\aleph_b$ represents the bias (initialised randomly), ∇_i is the corresponding input node weight, and *act* represents the activation function The fully connected layer uses the SoftMax activation function to determine the probabilities of the object labelling observed in the input image.

4. Results and Discussion

The developed algorithm for a vehicle- and motion-tracking prediction system is implemented in Python and the results are analysed for the proposed object, geolocation, and decision-making algorithms. The framework runs on a PC with an Intel(R) i7-8700 CPU @3.20GHz, NVIDIA GTX-2070 GPU (8 GB), and 16 GB RAM. In the test, 90% of the data is used for training and 10% is used for testing. The sample frames showing input and decision making are shown in Figure 5.

4.1. Dataset

The VisDrone2019 dataset was gathered at the Lab of Machine Learning and Data Mining by the AISKYEYE team at Tianjin University in China. The standard dataset involved 288 video clips designed by 10,209 static images and 261,908 frames, monitored by several drone-mounted cameras and covering an extensive range of features incorporating location (gathered from 14 diverse cities detached by thousands of kilometres in China), objects (vehicles, pedestrian, and bicycles, etc.), environment (country and urban), and density (sparse and crowded scenes). It should be noted that the dataset was gathered under several drone platforms (i.e., drones with diverse methods), in various situations, and under different lighting and weather conditions.

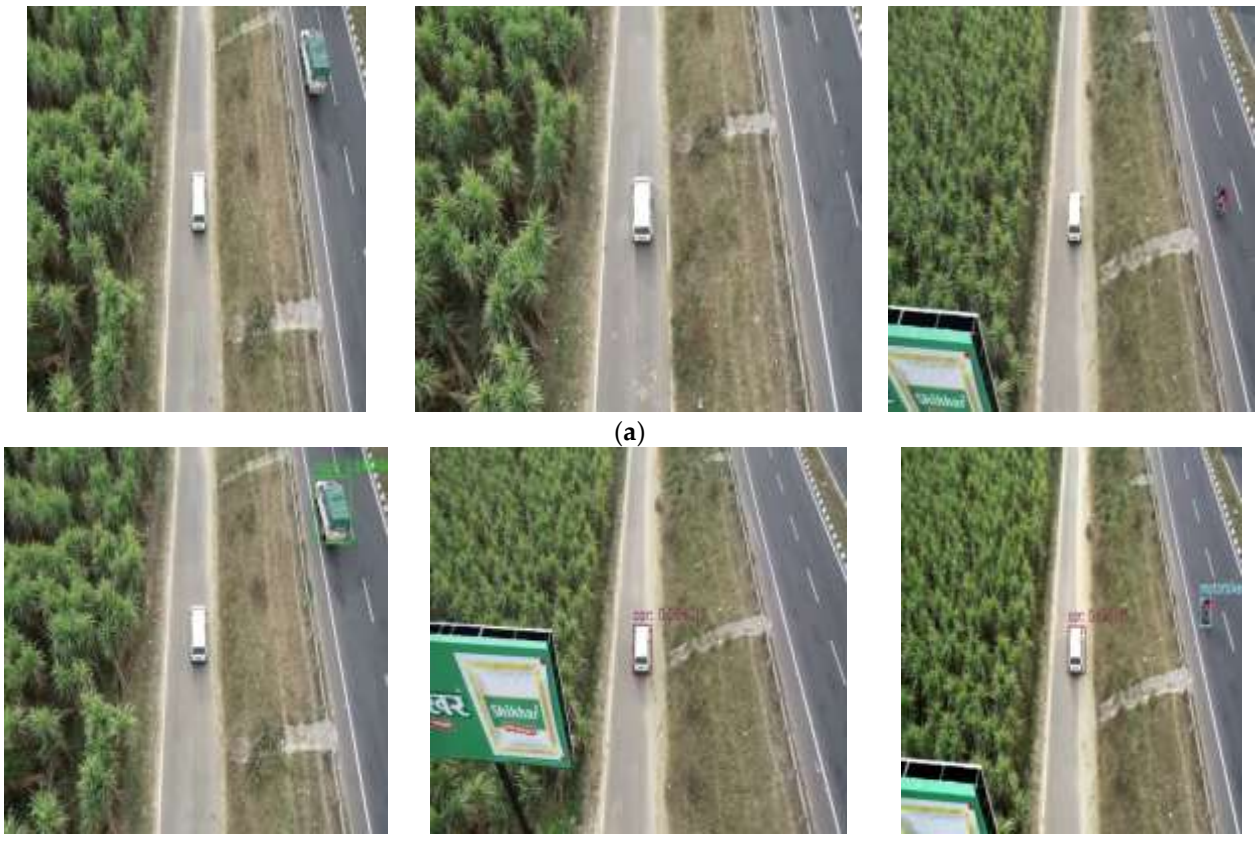


Figure 5. Proposed vehicle detection and motion detection sample frames. (**a**) Input image; and (**b**) detected image.

4.2. Performance Analysis of Proposed OSBMO for Object Detection

The proposed OSBMO object-detection technique is analysed based on Global Consistency Error, Rand Index, and Variation of Information. It is compared to the existing techniques such as Watershed, LevelSet, and Region Growing in order to determine whether the proposed scheme outperforms the current deep learning methods for object detection. The metrics on which the evaluation of the proposed method is performed are tabulated in Table 2.

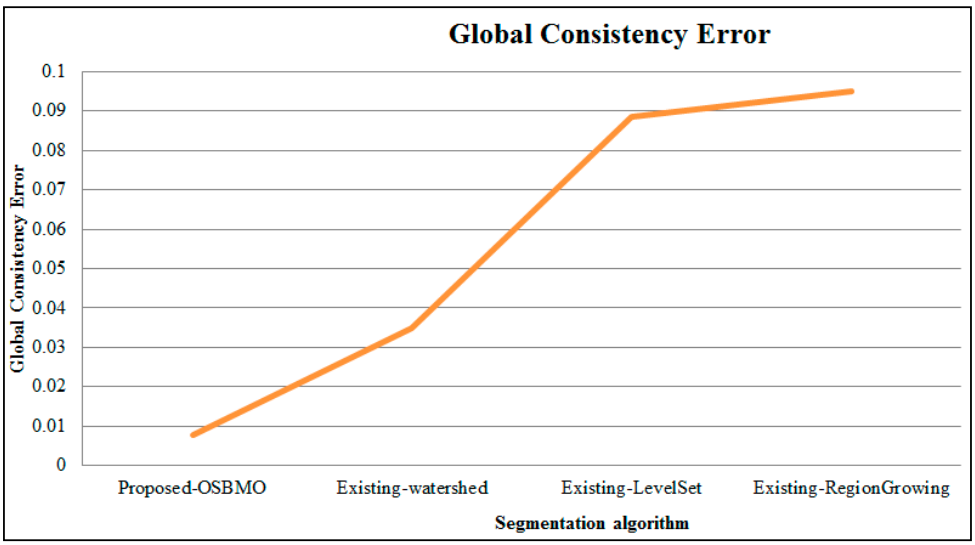
Table 2. Assessment of proposed OSBMO object-detection technique based on Global Consistency Error, Rand Index, and Variation of Information.

Method	Global Consistency Error	Rand Index	Variation of Information
Proposed: OSBMO	0.00789	0.95678	0.05883
Existing: Watershed	0.03482	0.89675	0.27554
Existing: LevelSet	0.08834	0.85678	0.11784
Existing: Region Growing	0.09499	0.81739	0.58831

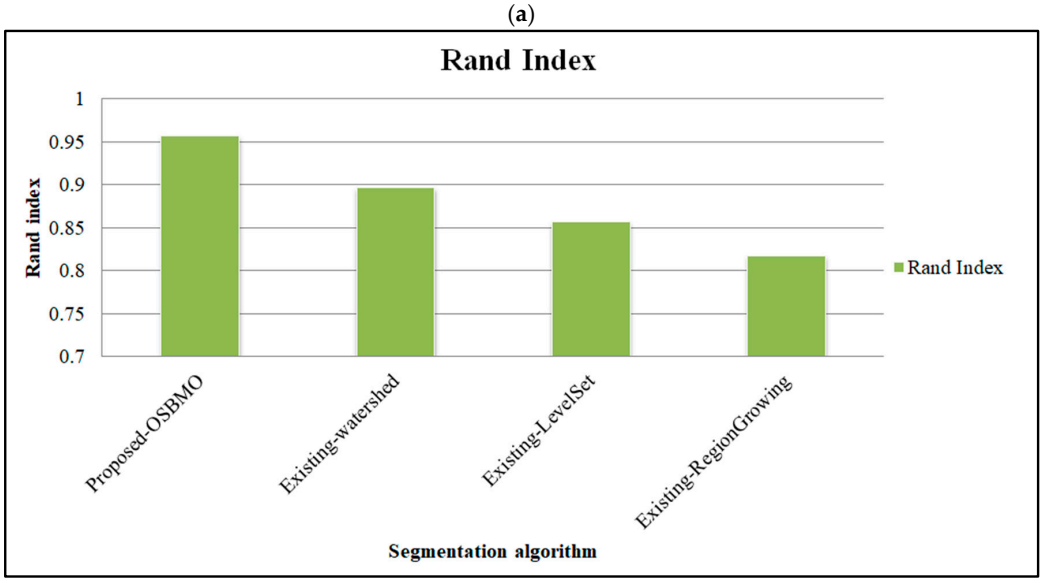
Table 2 represents the assessment of the proposed OSBMO object-detection strategy with different existing techniques in light of the measurements such as Global Consistency Error (GCE), Rand Index (RI), and Variation of Information (VOI). The assessment expresses the dependability of the proposed method in light of different datasets, uneven datasets, and delicate information, etc. The observed result shows that the proposed calculation will, in general, accomplish superior exhibition measurements values; for example, a GCR of

0.00789, RI of 0.95678, and a VOI value of 0.05883 which ranges between 0.00783–0.05883, though the current strategies accomplish a general presentation measurements value running between 0.034-0.89. The running value indicates that the proposed calculation ranges between the most minor and extreme values when contrasted with the current techniques. The proposed strategy will, in general, accomplish a better item identification output by limiting the blunder rate and covering intricacy when contrasted with current techniques. The graphical examination of the proposed work is illustrated in Figure 5.

Figure 6 represents the visual examination of the proposed calculation with different existing algorithms considering the presentation measurements. The measurements address the proficiency of the proposed strategy. The proposed OSBMO calculation accomplishes superior measurements values when contrasted with state-of-the-art methods. The proposed calculation distinguishes the item precisely, even in packed regions, with a lower error rate.

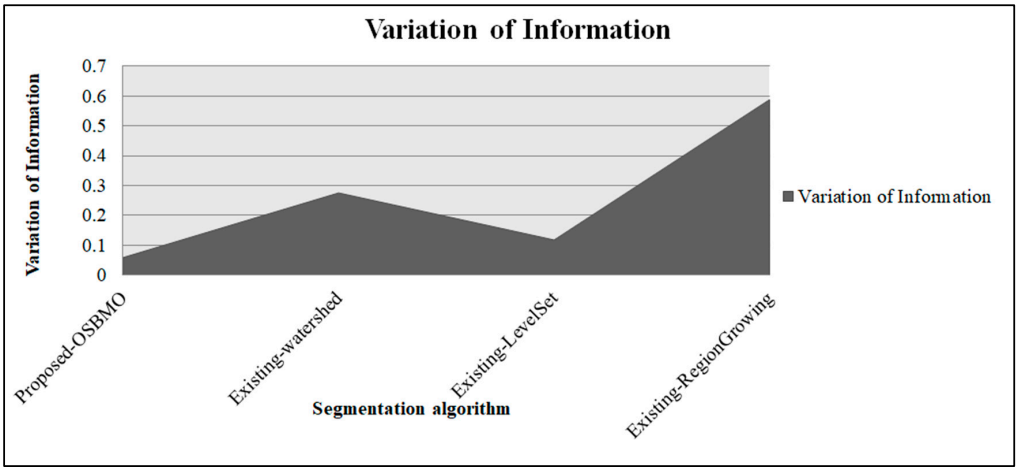


(**a**)



(**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. Graphical demonstration of proposed OSBMO algorithm regarding (**a**) Global consistency error; (**b**) Rand index; and (**c**) Variation of Information.

4.3. Performance Analysis of Proposed DBSCAN for Geolocation Motion Detection

In light of the measurements—for example, clustering error rate, bunching time, precision, and MSE—the proposed DBSCAN is examined with different existing methods. The examination is mainly completed to break down the geolocation of the articles inside various edges. The assessment of the measurements is arranged in Table 3.

Table 3. Performance analysis of Proposed DBSCAN based on clustering time for different frames.

No. of Frames	Object 1	Object 2	Object 3	Object 4	Object 5
100	38,705	26,314	21,754	19,474	9994
200	43,705	47,247	35,475	35,147	15,225
300	55,754	58,331	47,475	47,247	20,435
400	64,648	66,341	57,741	52,201	36,634
500	79,447	79,224	77,485	79,024	47,291

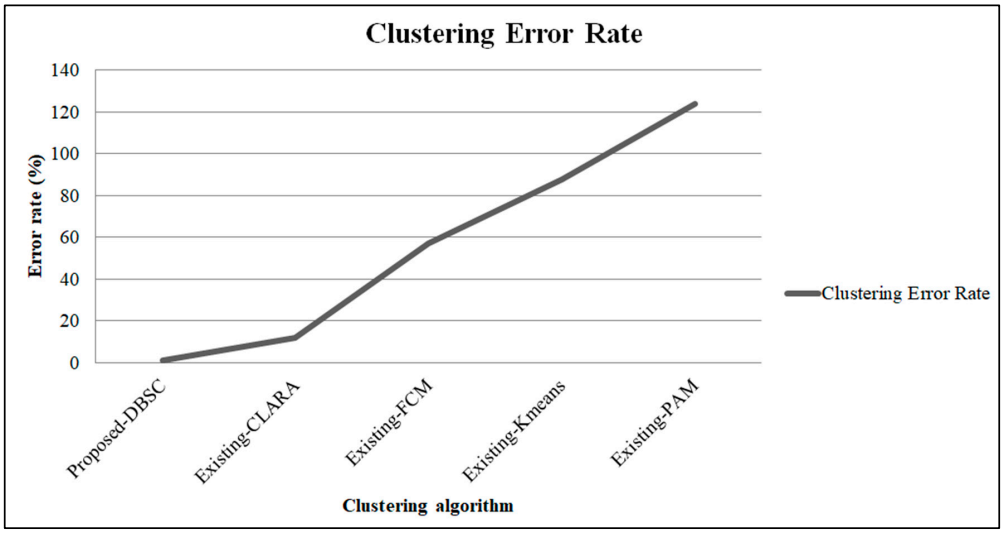
Table 3 delineates the clustering time taken to identify the target movement for various casings. The proposed DBSCAN procedure takes a grouping time running between 38,705 s and 79,447 s for Object 1 for outlines ranging from 100 to 500. The proposed strategy will generally perform in a generally comparable way for various items. The movement and geolocation following is achieved due to the Manhattan distance. Inside low time, the work is fit for accomplishing a high reaction movement following common clustering error and mean square error, as is displayed in Table 4.

Table 4. Performance analysis of Proposed DBSCAN based on clustering time, accuracy, and MSE.

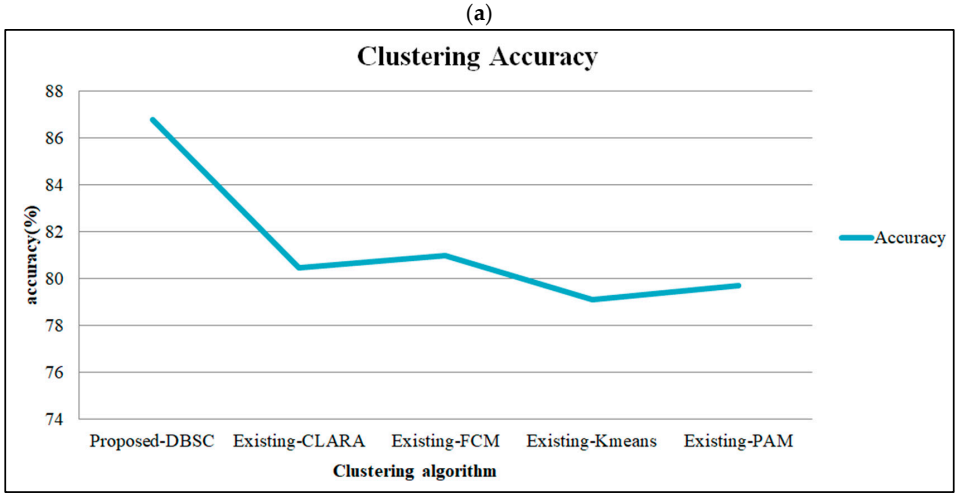
Method	Clustering Error Rate	Accuracy	Mean Squared Error
Proposed DBSC	0.82201	86.7866	0.58392
Existing CLARA	11.87632	80.4481	12.77381
Existing FCM	56.94459	80.9932	180.483
Existing K-means clustering	87.89034	79.1002	58.92581
Existing PAM	123.8477	79.6883	276.7011

Table 3 delineates the metric examination of the proposed DBSCAN with the existing strategies to assess movement detection. The misleading recognition emerges predominantly due to ill-advised algorithm learning, system intricacy, etc.; reasons for which there is a possibility of misdetection. The proposed DBSCAN, in general, accomplished a lower Clustering Error Rate of 0.822, MSE value of 0.58392, and a higher accuracy of 86.78%. However, the existing CLARA, FCM, K-Means Grouping, and PAM methods accomplish a higher benefit of clustering error rate, an MSE in the middle between 11.87 and 276.7, and a lower precision value ranging between 79.10% and 80.99%. Based on the observed measurement values, it may be expressed that the proposed strategy performs better than the existing techniques and will generally distinguish the geolocation of the articles for various objects more precisely. The graphical portrayal of the proposed calculation is represented in Figure 6.

Figure 7 graphically represents the clustering error rate, accuracy, and MSE metrics analysis of the proposed DBSCAN with various existing algorithms such as CLARA, FCM, K-Means Clustering, and PAM. The analysis states that the proposed method avoids false detection and obtains an accurate result compared to the existing techniques.

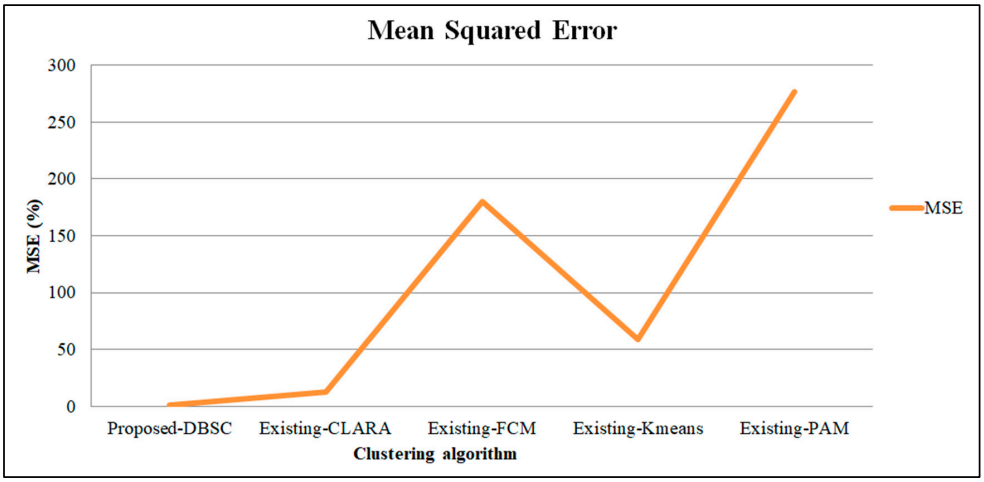


(**a**)



(**b**)

Figure 7. *Cont.*



(**c**)

Figure 7. Graphical demonstration of proposed DBSCAN regarding (**a**) error rate; (**b**) accuracy; and (**c**) mean square error.

4.4. Performance Analysis of Proposed RL for Object Detection

The proposed RL is inspected and fixated on the measurements such as accuracy, recall, precision, computation time, FPR, and FNR with various existent techniques such as YOLOv4, Resnet50, VGG19, and CNN. Table 5 arranges the proposed strategy's assessment along with the standard techniques focused on vehicle detection.

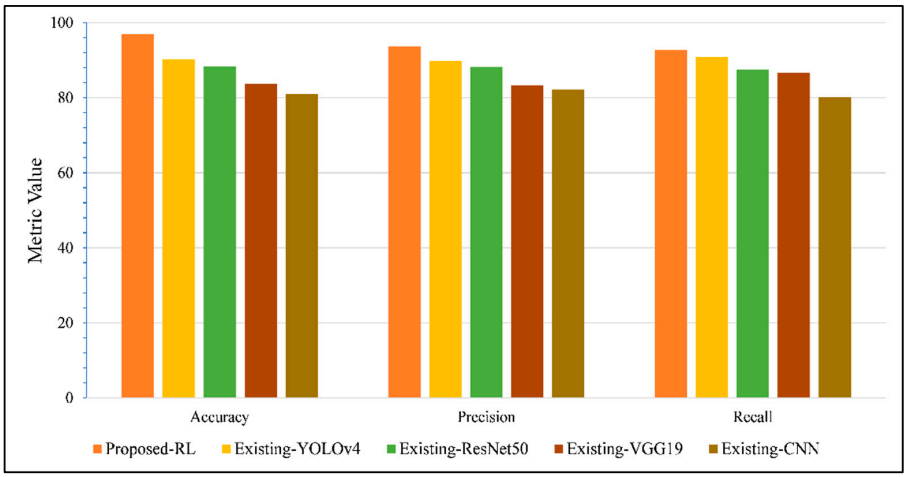
Table 5. Performance Analysis of Proposed RL Based on Classification metrics.

Method	Accuracy	Precision	Recall	FPR	FNR	Computation Time
Proposed: RL	96.8342	93.6754	92.6482	0.0646	0.03396	15,785
Existing: YOLOv4	90.1843	89.7643	90.7653	0.0939	0.90403	67,823
Existing: ResNet50	88.3834	88.2242	87.522	0.1736	0.78891	99,234
Existing: VGG19	83.7456	83.3146	86.7103	0.4829	0.99905	55,746
Existing: CNN	80.9933	82.1318	80.2295	0.8943	0.89932	89,003

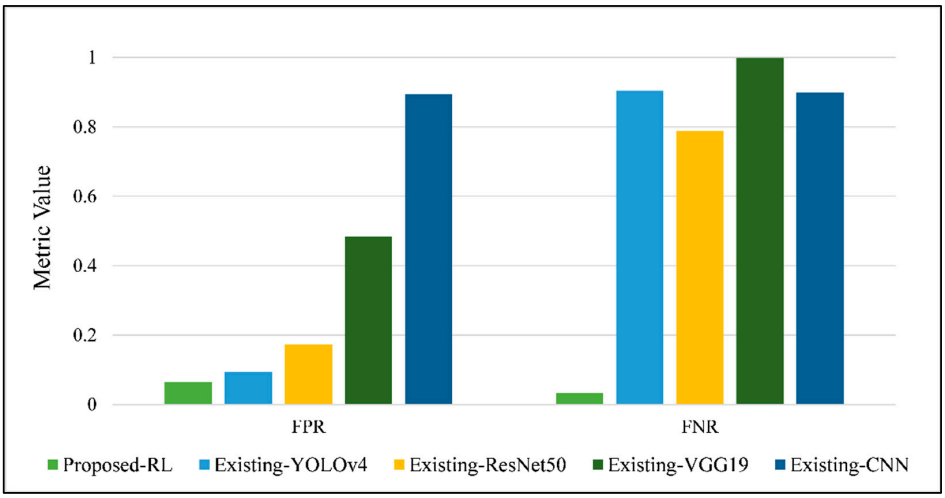
Table 5 displays the proposed RL along with various existing techniques such as YOLOv4, Resnet50, VGG19, and CNN, focusing on measurements, i.e., accuracy, recall, precision, computation time, FPR, and FNR. The four fundamental parameters such as true negative (TN), true positive (TP), false negative (FN), and false positive (FP), form the basis of the performance assessment metrics. The previously mentioned boundaries are the premise focused on the exhibition measurements. As a TP characterises that the real value is an object and the value anticipated matches the same; a TN describes that the genuine value is not an object and the value anticipated likewise yields something similar; an FP establishes that the actual value is not an object, yet the value anticipated is showing an object, and an FN characterises that the actual value is an object, but the predicted value is not detecting the object. Thus, an object-detection assessment is dependent on the four indices mentioned earlier. Indeed, the proposed algorithm is very robust and has overcome the previous algorithms. Benjdira et al. [25] compared the Fast R CNN with YOLOv3 for car detection in UAV images and concluded that the former was slower than YOLOv3.

However, our proposed algorithm based on FRCNN has surpassed the next version of YOLOv3.

Figure 8 shows the proposed RL along with the various existent techniques, such as YOLOv4, Resnet50, VGG19, and CNN, focused on measurements, i.e., accuracy, recall, precision, computation time, FPR, and FNR. The measurement accuracy and review symbolise the work's satisfactoriness on the assorted dataset, i.e., the proposed order strategies' unwavering quality. When compared to other state-of-the-art deep learning architectures, the proposed framework achieves a more remarkable accuracy, recall, precision, FPR, FNR, and computation time of 96.83%, 92.64%, 93.67%, 0.0646%, 0.0337%, and 15785s, respectively. In any case, the existent method accomplishes the metric value running in between 0.1736% and 90.76% that embodies a lesser plan viability as analogised to the procedure proposed. In addition, the proposed method is broken down and fixated on computation time measurements that depict the exactness of unbalanced dispersion probability. With respect to quantifying measures, the strategy presented yields a higher worth of object detection and tracking and avoids a false detection rate, leading to a low error rate. In this way, the proposed strategy yields proficient unwavering quality and sidesteps object detection as analogised to existing procedures.

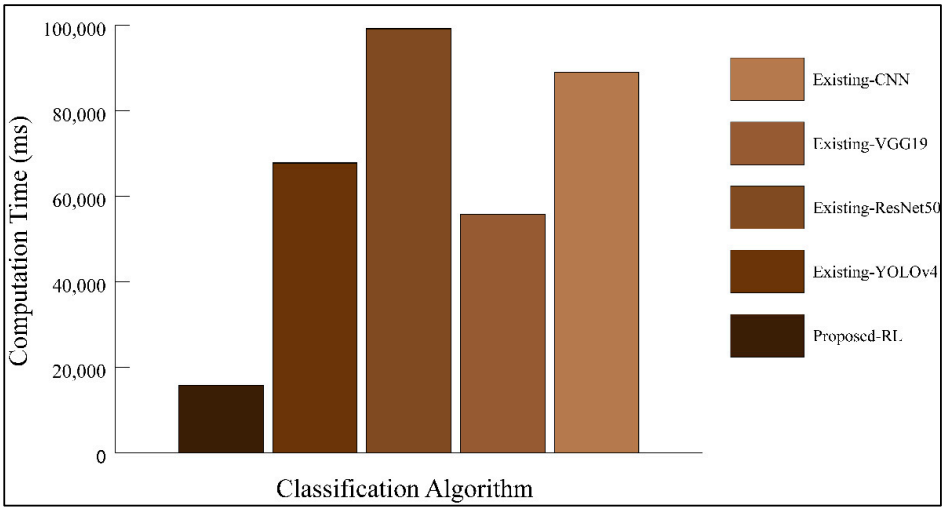


(**a**)



(**b**)

Figure 8. *Cont.*



(**c**)

Figure 8. Graphical demonstration of the proposed RL technique regarding the statistical analysis (**a**) Positive measure; (**b**) negative measure; and (**c**) computation time.

5. Conclusions

The proposed work introduced a challenging UAV benchmark containing various UAV recordings acquired in complex scenarios. To expand the cues, e.g., appearance and movement in following given UAV information, the work has proposed an intelligent, self-advanced, and consistent methodology for programmed vehicle identification, following and geolocation in UAS-acquired images that utilise recognitions, area and following highlights to upgrade an ultimate conclusion. A productive, ideal reinforcement learning calculation due to a Fast RCNN has made an ideal approach to precise vehicle identification. The proposed work performs well in a thickly filled parking garage, an intersection, a crowded street, and so on. The work handles identifying small objects by overcoming the problems of finding a reasonable element space and causing various vehicles to resemble each other.

Finally, the work was evaluated alongside several cutting-edge identification and following methodologies on the benchmark information with notable credits for UAVs. The trial results demonstrated how the proposed model could make the following outcomes more potent in both single- and numerous- object following. As the characteristics of UAV stage information are alterable in various conditions, the scene priors ought to be viewed in identification and following techniques.

Author Contributions: Conceptualisation, C.H.S., V.M.; methodology, C.H.S., V.M. and A.K.S.; validation, C.H.S., V.M.; formal analysis, C.H.S., V.M. and A.K.S.; writing—original draft preparation, C.H.S., V.M. and A.K.S.; writing—review and editing, C.H.S., V.M., K.J. and A.K.S.; supervision, K.J. and A.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study's findings are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank to the Indian Institute of Technology, Roorkee for the use of their resources. The authors are also thankful to the anonymous reviewers for their constructive suggestions for improving the quality of the present work.

Conflicts of Interest: This manuscript has not been published or presented elsewhere in part or in entirety and is not under consideration by another journal. There are no conflicts of interest to declare.

References

1. Kelechi, A.H.; Alsharif, M.H.; Oluwole, D.A.; Achimugu, P.; Ubadike, O.; Nebhen, J.; Aaron-Anthony, A.; Uthansakul, P. The Recent Advancement in Unmanned Aerial Vehicle Tracking Antenna: A Review. *Sensors* **2021**, *21*, 5662. [CrossRef] [PubMed]
2. Chen, R.; Li, X.; Li, S. A lightweight CNN model for refining moving vehicle detection from satellite videos. *IEEE Access* **2020**, *8*, 221897–221917. [CrossRef]
3. Chen, Y.; Zhao, D.; Er, M.J.; Zhuang, Y.; Hu, H. A novel vehicle tracking and speed estimation with varying UAV altitude and video resolution. *Int. J. Remote Sens.* **2021**, *42*, 4441–4466. [CrossRef]
4. Balamuralidhar, N.; Tilon, S.; Nex, F. MultEYE: Monitoring system for real-time vehicle detection, tracking and speed estimation from UAV imagery on edge-computing platforms. *Remote Sens.* **2021**, *13*, 573. [CrossRef]
5. Butilă, E.V.; Boboc, R.G. Urban Traffic Monitoring and Analysis Using Unmanned Aerial Vehicles (UAVs): A Systematic Literature Review. *Remote Sens.* **2022**, *14*, 620. [CrossRef]
6. Shan, D.; Lei, T.; Yin, X.; Luo, Q.; Gong, L. Extracting key traffic parameters from UAV video with on-board vehicle data validation. *Sensors* **2021**, *21*, 5620. [CrossRef]
7. Zhou, W.; Liu, Z.; Li, J.; Xu, X.; Shen, L. Multi-target tracking for unmanned aerial vehicle swarms using deep reinforcement learning. *Neurocomputing* **2021**, *466*, 285–297. [CrossRef]
8. Byun, S.; Shin, I.-K.; Moon, J.; Kang, J.; Choi, S.-I. Road traffic monitoring from UAV images using deep learning networks. *Remote Sens.* **2021**, *13*, 4027. [CrossRef]
9. Srivastava, S.; Narayan, S.; Mittal, S. A survey of deep learning techniques for vehicle detection from UAV images. *J. Syst. Archit.* **2021**, *117*, 102152. [CrossRef]
10. Darehnaei, Z.G.; Fatemi, S.M.J.R.; Mirhassani, S.M.; Fouladian, M. Ensemble Deep Learning Using Faster R-CNN and Genetic Algorithm for Vehicle Detection in UAV Images. *IETE J. Res.* **2021**, 1–10. [CrossRef]
11. Abdelmalek, B.; Zarzour, H.; Kechida, A.; Taberkit, A.M. Vehicle detection from UAV imagery with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6047–6067.
12. Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 91–124. [CrossRef]
13. Avola, D.; Cinque, L.; Diko, A.; Fagioli, A.; Foresti, G.; Mecca, A.; Pannone, D.; Piciarelli, C. MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images. *Remote Sens.* **2021**, *13*, 1670. [CrossRef]
14. Memon, S.A.; Ullah, I. Detection and tracking of the trajectories of dynamic UAVs in restricted and cluttered environment. *Expert Syst. Appl.* **2021**, *183*, 115309. [CrossRef]
15. Xin, L.; Zhang, Z. A vision-based target detection, tracking, and positioning algorithm for unmanned aerial vehicle. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5565589.
16. Boudjit, K.; Ramzan, N. Human detection based on deep learning YOLO-v2 for real-time UAV applications. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 527–544. [CrossRef]
17. Zhao, X.; Pu, F.; Wang, Z.; Chen, H.; Xu, Z. Detection, tracking, and geolocation of moving vehicle from uav using monocular camera. *IEEE Access* **2019**, *7*, 101160–101170. [CrossRef]
18. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Valappil, N.K.; Memon, Q.A. CNN-SVM based vehicle detection for UAV platform. *Int. J. Hybrid Intell. Syst.* **2021**, *preprint*.
20. Li, B.; Yang, Z.-P.; Chen, D.-Q.; Liang, S.-Y.; Ma, H. Maneuvering target tracking of UAV based on MN-DDPG and transfer learning. *Def. Technol.* **2021**, *17*, 457–466. [CrossRef]
21. Espsoito, N.; Fontana, U.; D'Autilia, G.; Bianchi, L.; Alibani, M.; Pollini, L. A hybrid approach to detection and tracking of unmanned aerial vehicles. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 1345.
22. Shao, Y.; Mei, Y.; Chu, H.; Chang, Z.; Jing, Q.; Huang, Q.; Zhan, H.; Rao, Y. Using Multiscale Infrared Optical Flow-based Crowd motion estimation for Autonomous Monitoring UAV. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 589–593.
23. Quanfu, F.; Brown, L.; Smith, J. A closer look at Faster R-CNN for vehicle detection. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gotenburg, Sweden, 19–22 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 124–129.
24. Sutton, R.S.; Precup, D.; Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.* **1999**, *1*, 181–211. [CrossRef]
25. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.