Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040738

ARTICLE

Intelligent Traffic Surveillance through Multi-Label Semantic Segmentation and Filter-Based Tracking

Asifa Mehmood Qureshi[1], Nouf Abdullah Almujally[2], Saud S. Alotaibi[3], Mohammed Hamad Alatiyyah[4] and Jeongmin Park[5,*]

[1]Department of Creative Technologies, Air University, Islamabad, 44000, Pakistan

[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[3]Information Systems Department, Umm Al-Qura University, Makkah, Saudi Arabia

[4]Department of Computer Science, College of Sciences and Humanities in Aflaj, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

[5]Department of Computer Engineering, Tech University of Korea, Siheung-si, Gyeonggi-do, 15073, Korea

*Corresponding Author: Jeongmin Park. Email: jmpark@tukorea.ac.kr

Received: 29 March 2023 Accepted: 04 July 2023 Published: 08 October 2023

ABSTRACT

Road congestion, air pollution, and accident rates have all increased as a result of rising traffic density and worldwide population growth. Over the past ten years, the total number of automobiles has increased significantly over the world. In this paper, a novel method for intelligent traffic surveillance is presented. The proposed model is based on multilabel semantic segmentation using a random forest classifier which classifies the images into five classes. To improve the results, mean-shift clustering was applied to the segmented images. Afterward, the pixels given the label for the vehicle were extracted and blob detection was applied to mark each vehicle. For the validation of each detection, a vehicle verification method based on the structural similarity index is proposed. The tracking of vehicles across the image frames is done using the Identifier (ID) assignment technique and particle filter. Also, vehicle counting in each frame along with trajectory estimation was done for each object. Our proposed system demonstrated a remarkable vehicle detection rate of 0.83 over Vehicle Aerial Imaging from Drone (VAID), 0.86 over AU-AIR, and 0.75 over the Unmanned Aerial Vehicle Benchmark Object Detection and Tracking (UAVDT) dataset during the experimental evaluation. The proposed system can be used for several purposes, such as vehicle identification in traffic, traffic density estimation at intersections, and traffic congestion sensing on a road.

KEYWORDS

Traffic surveillance; multi-label segmentation; random forest; particle filter; computer vision

1 Introduction

For numerous real-time computer vision techniques, fast image frame sequence processing is crucial. One of the most significant areas is the tracking of moving objects in video image sequences such as traffic control and surveillance, sports reporting, and video annotation [1]. The number of

vehicles has increased drastically over the past few years. Therefore, there is a need to automate the traffic surveillance systems. A large number of image-based systems have been proposed by the research community. But there are still some challenges that need to be addressed to enhance the traffic monitoring system capabilities. A large number of effective image processing techniques have been proposed which perform well for static image data. However, these scenarios will get more challenging if the background and moving objects change dynamically [2]. Techniques including background subtraction, and consecutive frame differencing are not suitable when the images are captured using a mobile platform because background pixels also have motion in them which classifies them as foreground objects. Therefore, several areas of computer vision and image processing, including intelligent transportation, medical images, object recognition, semantic segmentation, and human-computer interaction, have proven to be quite effective [3]. Semantic segmentation is the grouping and individual labeling of pixels that belong to the same class [4]. Traditional traffic monitoring systems only consist of binary segmentation e.g., vehicle and background labeling. However, our proposed system performs multi-class segmentation for a better understanding of the scene and different objects. Furthermore, Aerial data has the potential to greatly improve traffic management, control, efficiency, and effectiveness. But it also has some challenges which include varying object size, large covered areas other than the roads, and different road designs which need to be addressed effectively to develop systems based on data retrieved from mobile platforms.

This paper proposes a reliable system for traffic monitoring in aerial images specifically designed by keeping in view the above-mentioned limitations. The approach requires segmenting all Red, Green and Blue (RGB) images into various classes which include vehicles, roads, buildings, sky, and greenery. Then, to further improve the result the segmented images are subjected to mean-shift clustering to group the pixels having the same class labels. The phase of vehicle detection comes next, which consists of two steps i) extracting only those pixels that belong to the vehicle class and ii) finding contour by detecting the borders of the object. To verify each detected vehicle the Structural Similarity Index (SSID) score was calculated using each image's corresponding mask. Afterward, the traffic densities on the road were estimated by counting each verified vehicle. To track multiple vehicles within a single frame a unique ID was allocated founded on a distinctive feature descriptor named Oriented FAST Features from Accelerated Segment Test (ORB). Finally, the location of each vehicle was estimated by using the particle filter, and also the allocated IDs were retrieved in each succeeding frame by matching the ORB key point descriptors. Vehicle trajectories were estimated for each tracked vehicle. Three large publicly available datasets, the UAVDT Dataset, AUAIR Dataset, and VAID dataset were used for experimentation purposes.

This paper's primary contributions are as follows:

- Multi-label pixel segmentation technique for accurate vehicle extraction from Red, Green, and Blue (RGB) images.
- Proposing an easy and efficient way for detection verification based on SSID score using ground truth.
- Designing a powerful vehicle recognition system grounded on the ORB features for ID retrieval and particle filter for tracking.

The rest of the paper is structured as follows: Section 2 explains and evaluates the research work that is pertinent to the proposed system. Section 3 defines the overall system methodology. The dataset used in the proposed work is described in Section 4, which also uses several tests to demonstrate the system's robustness. The research is concluded in Section 5, which also lists some future directions.

2 Related Work

Researchers have been actively working on traffic monitoring algorithms for the past few years. They investigated their systems' behaviors using images taken from a static camera, satellite images, and aerial images. In maximum cases, the whole images are firstly preprocessed to remove irrelevant areas other than vehicles and then features are extracted from them. Different approaches are based on image differencing, foreground extraction, or background subtraction techniques. These approaches are simple and especially useful when the Region of Interest (ROI) is visible and of reasonable size in images [5]. However, in aerial images, the vehicle size varies depending on the height at which the images are taken. Therefore, semantic segmentation approaches are being used for detection and tracking purposes [6]. Moreover, the additional steps of clustering and identifier assignment for improved results are also common. Thus, the related work is categorized into semantic-based and deep learning-based traffic monitoring systems to present an overview of existing models and techniques.

2.1 Semantic Segmentation-Based Traffic Monitoring Systems

Zhang et al. [7] have performed aerial vehicle recognition by deploying a multi-label semantic segmentation mechanism for better scene understanding. They used Mask Region-based Convolutional Neural Network (R-CNN) to segment different regions and then eliminate background objects to reduce the computational area. To detect the aerial vehicle, visual attention mechanism was used for feature extraction which was passed onto the Adaboost classifier to get the exact location. Makrigiorgis et al. [8] incorporated segmentation for road extraction using EfficientNet which combines MobileNetV2 and ResNet18. Further, the You Look Only Once version 3 (YOLOv3) algorithm detects the vehicles on the extracted ROI. In complex cases, background elimination in real-time scenarios is more challenging. Also, after the removal of invalid data deploying a pre-trained deep learning algorithm only increase the computational complexity of the model as these models can perform well if directly applied to raw images. Their road extraction mechanism can be replaced by multi-label scene segmentation to better analyze the images and to directly get vehicles for detection.

Gomaa et al. [9] argued that in aerial images both the background and foreground are moving therefore approaches based on detecting motion are not feasible. Thus, a method based on top-hat and bottom-hat transformation along with the Otsu partitioning method and morphological operations were deployed for detection. While vehicle motion is important, Shi Tomasi features were extracted and clusters grounded on displacement and angle trajectories were formed. The background clusters were removed, leaving behind the vehicles. Robust features of each vehicle were used for tracking across images. They achieved high accuracy by using multiple feature maps. In another study [10], an object detection method for images taken under low-illumination conditions has been proposed. The methodology presented a two-stage approach i.e., cloud-based image enhancement and edge-based detection which is an efficient and dynamic approach to address each image contrast enhancement requirement separately. The authors of [11] employed an innovative method for image stacking. Only small cars were included in the image registration procedure, and all of the stationary backgrounds near the moving vehicles were blurred using the warping technique. This algorithm's primary objective is to eliminate distracting image background elements that can be smoothed to extract only the vehicle from the surrounding area. These methods, however, were distinguished by complex features and these systems have high time complexity.

2.2 Deep Learning-Based Traffic Monitoring Systems

Numerous researchers have implemented a feature detection approach for directly recognizing vehicles in images. Kong et al. [12] used a vehicle detection technique based on salient point feature extraction for image stabilization. A particle filter using a Histogram of Gradient (HOG) features was used for tracking across frames. Gupta et al. [13] deployed different deep-learning models directly to images to detect vehicles. The models include a two-stage detector named Faster Recurrent Convolutional Neural Network (R-CNN) in comparison with one-step detectors i.e., Single Shot Detector (SSD), YOLOv3, and YOLOv4. YOLOv4 algorithm outperforms all other models by having an 88% mean average precision (mAP) score. These models are highly sensitive to class imbalance and therefore require data-augmenting methodologies. Ozturk et al. [14] proposed a vehicle detection model primarily using Convolutional Neural Network (CNN) with the support of morphological corrections named miniature CNN architecture. This post-processing is computationally expensive. Additionally, alternative datasets of aerial images do not exhibit the same accuracy. The combination of deep learning for feature extraction and Support Vector Machines (SVMs) for classification is described in [15]. This method's use of a brute-force search methodology results in a higher computing intensity.

Baykara et al. [16] used the YOLO method to find the vehicles. Lane polygon and lane ID detection were used for vehicle tracking. A vehicle is passed to the tracking module, which gives each newly discovered vehicle an ID when its centroid falls within the lane polygon. In [17], the detection of moving vehicles was done using a frame differencing method. While CNN was utilized for classification, the Kalman filter was used to further track the vehicles.

3 Proposed Method

This section elaborates on the proposed traffic monitoring system. An overview of the system architecture is shown in Fig. 1. All RGB images were segmented into 5 classes using the Random Forest classifier. To smoothen the obtained segmented images and to reduce noise, mean-shift clustering was applied to make clusters of the pixels having identical class labels. For vehicle detection, first of all, the pixels which belong to the vehicle class were extracted and then contours were detected by detecting each object's edges. To verify each segmented vehicle, the SSID score was calculated using the image masks (ground truth). The density of traffic on the road was estimated by counting each verified vehicle. To track multiple vehicles, a unique ID was allocated based on the ORB features. Vehicles were tracked across multiple image frames using a particle filter. To locate each tracked vehicle, IDs were restored based on ORB key point descriptor along with trajectories approximation. The different stages of the proposed framework are thoroughly explained in the following subsections

3.1 Image Pre-Processing

Firstly, the RGB images from all three datasets are cropped to a constant dimension of 300×300 to maintain consistency in size. Then, these images were converted to grayscale to reduce the number of channels.

3.2 Image Segmentation

Image segmentation is used as a fundamental stage in many visual technologies that aim to assess situations [18]. As a result, segmentation plays a crucial part in numerous applications, such as surveillance systems, driverless vehicles, virtual reality, and medical imaging. Researchers have

developed a variety of object segmentation algorithms [19], including watershed transform [20], region-growing, graph-cuts, k-means clustering [21], conditional random fields, and more sophisticated deep learning (DL) techniques [22]. To find the best solution for the segmentation of traffic scenes we used a random forest classifier which outperforms other classifiers. To train the model, different features were extracted from the images. These feature sets were then split into which includes the original pixel values, Gabor filter, Scharr filter, Prewitt filter, gaussian filter, median filter, and variance [23]. These features were based on the edges and the color space changes which helped detect different regions in the images.

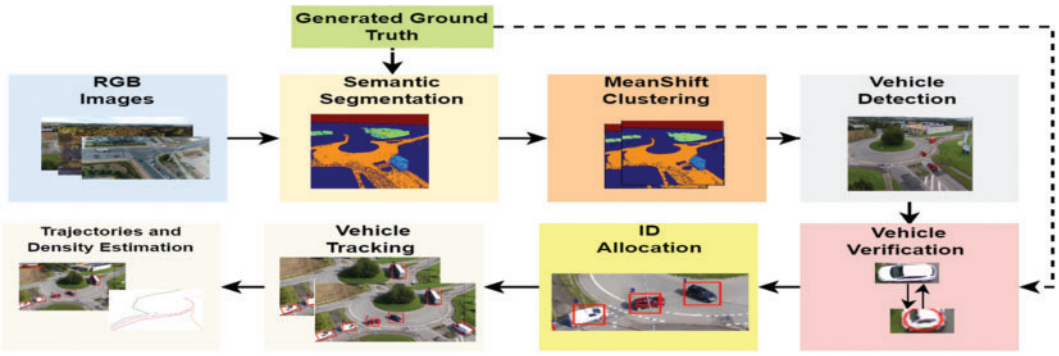


Figure 1: An overview of the proposed intelligent traffic surveillance system

First of all, the pixel value of the original image was taken as feature 1. Then, the Gabor filter was applied which is a linear filter used for disparity estimation, feature extraction, texture categorization, and edge detection. The Gabor kernel can be expressed as Eq. (1).

$$g\left(x, y; \lambda, \theta, \psi, \sigma, \gamma\right) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left[i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right] \tag{1}$$

where $x' = -x\ sin\ \theta + y\ cos\ \theta$ and $y' = x\ cos\ \theta + y\ sin\ \theta$ and x and y are the image coordinates. θ represents the parallel stripes direction of the filter, σ represents the standard deviation of the Gaussian component, γ identifies the aspect ratio determining the function support's ellipticity, and ψ denotes the phase of the plane wave.

The resultant matrix $R(x, y)$ is obtained by convolving the original image $I(x, y)$ with the Gabor filter, using Eq. (2).

$$R\left(x, y; \lambda, \theta, \psi, \sigma, \gamma\right) = \sum_{x'} \sum_{y'} I\left(x - x', y - y'\right) g\left(x', y'; \lambda, \theta, \psi, \sigma, \gamma\right) \tag{2}$$

Also, Scharr and Prewitt filters were applied to detect edges both in the horizontal and vertical direction and to highlight gradient edges using the first derivative. The magnitude and orientation of gradient using Eqs. (3) and (4).

$$|I| = \sqrt{I_x^2 + I_y^2} \tag{3}$$

$$\theta = tan^{-1}\left(\frac{Il_y}{ls_x}\right) \tag{4}$$

To extract features after reducing noise in the image a low pass filter Gaussian was applied whose kernel is computed by using Eq. (5).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{\kappa^2+y^2}{2\sigma^2}} \tag{5}$$

where G is the Gaussian kernel. To have an extensive and meaningful feature vector, a median filter was also applied to remove salt and pepper noise as it replaces every pixel value with the median value. As the task was to multi-label the image therefore to measure the deviation of each pixel value from its mean value, the variance was also computed by using Eq. (6).

$$S^2 = \frac{\sum (x_i - \overline{x})}{n - 1} \tag{6}$$

where S is the variance, $\overline{x}$ is the mean. The final feature vector is of size 90000 × 7. The result of multi-label semantic segmentation can be visualized in Fig. 2. The images are segmented into five classes i.e., sky, buildings, vehicles, ground and road.

Figure 2: Output of image segmentation. (a) Original image, (b) after semantic segmentation

3.3 MeanShift Clustering

To further improve the segmentation accuracy of each class and to remove noise we applied mean shift clustering. It is a gradient ascent approach to calculate the local greatest density of a data collection by applying mean shifts. It is a non-parametric method that works well to find clusters in the data with arbitrary shapes [24]. The fundamental form of the x mean shift vector can be calculated using Eq. (7) under the presumption that n sample points in x_i, where i = 1, 2, . . . , n, is given in the d-dimensional space R^d.

$$M(x)_h = \frac{1}{k} \sum_{x_i \in s_h} (x_i - x) \tag{7}$$

where h denotes the radius and s_h represents the high-dimensional spherical area, satisfying the y-point set relationship using Eq. (8).

$$D_h(x) = \left\{ y : (y - x)^T (y - x) \leq h^2 \right\} \tag{8}$$

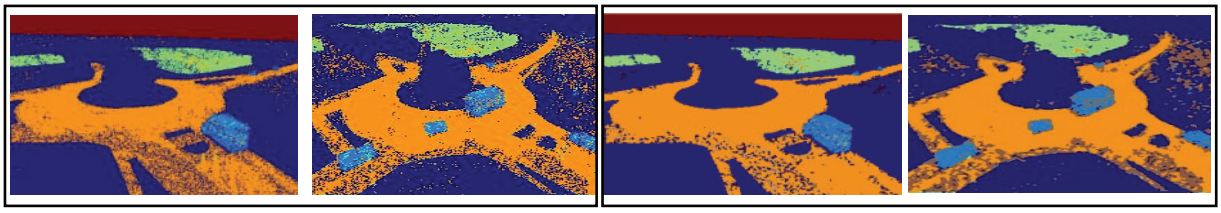
where k denotes that k points in x_i fall within the boundaries of D_h. Two elements, notably the neighborhood and color pixel bandwidths, have an impact on the mean shift method's final clustering. For the x_i points that fall within the bounds of D_h, the following rules are defined.

When the pixel bandwidth is short, the probability density is high when comparing the colors of pixels x and x_i. When comparing the distances of the pixels x and x_i, the high probability density is

shown by small distance bandwidth between x and x_i. As a result, these two rules can be combined to form the probability density function. Thus, the kernel function can be defined by using Eq. (9).

$$K_{h_s^2,h_r^2}(x) = \frac{C}{h_s^2 h_r^2} K\left(||\frac{x^s - x_i^s}{h_s}||^2\right) K\left(||\frac{x^r - x_i^r}{h_r}||^2\right) \tag{9}$$

where K(·) represents the kernel function. h_s denotes the distance bandwidth. h_r represents the color bandwidth, $K\left(||\frac{x^s - x_i^s}{h_s}||^2\right)$ reflects the space location of information, color information is represented by $K\left(||\frac{x^r - x_i^r}{h_r}||^2\right)$ and $\frac{C}{h_s^2 h_r^2}$ signifies unit density. The output of the segmentation process is shown in Fig. 3.



(a) (b)

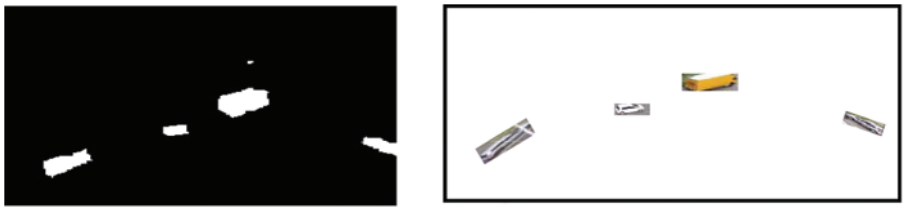
Figure 3: Result of mean-shift clustering. (a) Segmentation with noise, (b) mean-shift clustering

3.4 Vehicle Detection

Following multi-class semantic segmentation, we extract only those pixels that belong to the vehicle class as each pixel is tagged and allocated to a certain class during segmentation. As we only wanted to detect vehicles, we set all of the pixels' values to zero except for the vehicle class. After extracting the pixels of the vehicle class, the resultant image was converted into a binary image using Eq. (10).

$$bw(x,y) = \begin{Bmatrix} 1 \; if \; l(x,y) > 0 \\ 0 \; if \; (x,y) \leq 0 \end{Bmatrix} \tag{10}$$

where L stands for an image that only contains pixels of the vehicle class, and *bw* stands for the overall binary image that results, as seen in Fig. 4.



(a) (b)

Figure 4: Vehicle pixel extraction. (a) Binary image with only vehicle masks (b) the resulting image

As the extracted vehicle differs in color or brightness from the surroundings, therefore, to identify each vehicle separately a blob detection technique [25] was applied as represented in Fig. 5.

Figure 5: Vehicle detection using blob detection algorithm over AU-AIR, VAID, and UAVDT datasets

3.5 Vehicle Verification

To verify each segmented and detected vehicle, we back-propagate toward the ground truth of each particular image to confirm the presence of vehicles at certain locations. For verification, the Region of Interest (ROI) of each detection in segmented images, as well as ground truth, were extracted. Afterward, a Similarity Structure Index Measure (SSIM) was calculated to measure the similarity score between the vehicles [26]. SSIM consists of three key features of the image i.e., contrast, luminance, and structure as calculated by using Eqs. (11)–(13).

$$L(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{11}$$

$$C(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{12}$$

$$S(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_1} \tag{13}$$

where L denotes luminance, C denotes contrast and S denotes structure. μ_x and μ_y represent the sample mean of x and y. σ_x and σ_y are the standard deviation and σ_{xy} denotes the sample correlation coefficient between x and y. C_1 and C_2 are the constant need to stabilize the algorithm when the denominator approaches zero. Thus, the general formula of SSIM can be represented by using Eq. (14).

$$\text{SSIM} = I(x,y)^{\alpha} \cdot C(x,y)^{\beta} \cdot S(x,y)^{\gamma} \tag{14}$$

where α, β, and γ describe the relative importance of each feature. If the SSIM score is greater than 0.2 then the vehicle is added to the true positive. The proposed algorithm for vehicle verification is given in Algorithm 1.

Algorithm 1: Vehicle Detection and Verification

Input: S: segmented images, GT: Ground Truth, O: original image
Output: V*erifiedDetections* (v*1, v2, v3 . . . vn*)
Detections←[]
vehiclesilhouette←pixels having class label
keypoints← BlobDetection (*vehiclesilhouette*)
for all keypoint in keypoints **do**
    x←keypoint[0]
    y←keypoint[1]
    s←keypoint_size
    Detection←DrawRectangle (O, x, y, s)
V*erifiedDetections*←[]

(Continued)

Algorithm 1 (continued)

for *i* in range (len(*Detections*)):
ROI ← Getrectangularpatch (*Detections*[i])
ROI_GT← Getrectangularpatch (GT[i])
Score=SSIM (*ROI,ROI_GT*)
If score>0.2
V*erifiedDetections*←[S[i]]
end
end
return *VerifiedDetections* (v*1, v2, v3 . . . vn*)

3.6 ID Assignment Based on ORB Features

Each detected vehicle was given an ID based on ORB features before tracking to reidentify it in the following frames of the image.

ORB is a fast and efficient feature detector [27]. For key point detection, it makes use of the FAST (Features from Accelerated Segment Test) keypoint detector. It is an advanced form of the descriptor BRIEF (Binary Robust Independent Elementary Features). Also, it is invariant to scale and rotation. A patch moment is obtained by using Eq. (3).

$$m_{pq} = \sum x^p y^q I(x, y) \tag{15}$$

where p and q are the intensity values of the image pixels at x and y locations, respectively. Eq. (4) can be used to determine the center of mass using these moments.

$$C = \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \tag{16}$$

The patch orientation can be defined by Eq. (5).

$$\theta = \text{atan}(m_{01}, m_{10}) \tag{17}$$

The extracted ORB features were used to find the matching of tracked vehicles in the succeeding frames and if a matching is found the ID was restored otherwise the vehicle was registered with a new ID in the system. Fig. 6 shows the outcomes of applying an ORB feature description to the extracted vehicles and ID restoration throughout frames.

3.7 Vehicle Tracking

To track multiple vehicles across different image frames, the particle filter was applied. Particle filters are part of a broad class of Sequential Monte Carlo (SMC) techniques that are frequently utilized in tracking objects. To determine the minimum cost function, particle filters frequently start with the premise that the data distribution is unknown and that distribution "particles" or samples are assessed, examined, and aggregated into more meaningful conclusions [28]. For tracking, the posterior probability density at the t instant is estimated, which is acquired in the following two steps.

Step 1 Prediction: Assume that the posterior probability density function $p(g_{t-1} | o_{1:t-1})$ and starting probability density value $p(g_0)$ of the probability density are both known at the time $t-1$. $\boldsymbol{g_t}$ defines a three-dimensional vector where $\boldsymbol{g_t} = [g_t{}^x, g_t{}^y, g_t{}^s]$. The position of the object is expressed by $g_t{}^x, g_t{}^y$

whereas the change in size is represented by g_t^s. Thus, the prior probability can be defined using Eq. (18).

$$p(g_{t-1} \mid o_{1:t-1}) = \int p(g_t \mid g_{t-1}) p(g_{t-1} \mid o_{1:t-1}) dg_{t-1} \tag{18}$$

where $p(g_t \mid g_{t-1})$ represent the state equation of the target.

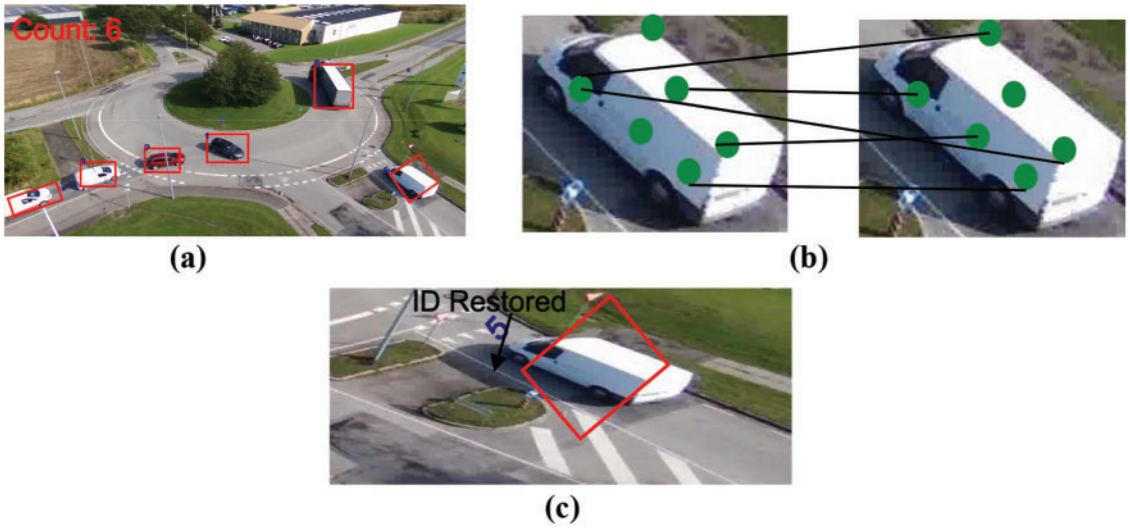


Figure 6: ID assignment and restoration (a) ID assigned to each vehicle based on ORB features (b) features matching across frames (c) ID restored for the same vehicle in succeeding frame

Step 2 Updating: the observation model of the system yields $p(g_t \mid o_{1:t})$ as given in Eq. (19).

$$p\left(g_t \mid o_{1:t}\right) = \frac{p(o_t \mid g_t) p(g_t \mid o_{1:t-1})}{p(o_t \mid g_{1:t-1})} \tag{19}$$

$p(o_t \mid g_t)$ represent the observation likelihood function which is obtained by the observation of the tracked object. Whereas, $(o_t \mid g_{1:t-1})$ is defined as a normalized constant. The recursive Bayesian filtering also known as particle filter is simulated by the non-parametric Monte Carlo Method as given in Eq. (20).

$$p\left(g_t \mid o_{1:t}\right) \approx \sum_{i=1}^{N} w_t^i \delta\left(g_t - g_t^i\right) \tag{20}$$

where w_t^i is the corresponding particle's weight. This weight is updated by using Eq. (21).

$$w_t^i = w_{t-1}^i \frac{p(o_t \mid g_t^i) p(g_t^i \mid g_{t-1}^i)}{q(g_t^i \mid g_{t-1}^i, o_t)}, \ \sum_{i=1}^{N} w_t^i = 1 \tag{21}$$

where $q(g_t^i \mid g_{t-1}^i, o_t)$ represent the proposed distribution function in Bayesian sampling. Finally, the estimated possible location of the target is obtained as presented in Eq. (22).

$$\hat{g}_t = \sum_{i=1}^{N} g_t^i w_t^i \tag{22}$$

The result of vehicle tracking can be seen in Fig. 7.

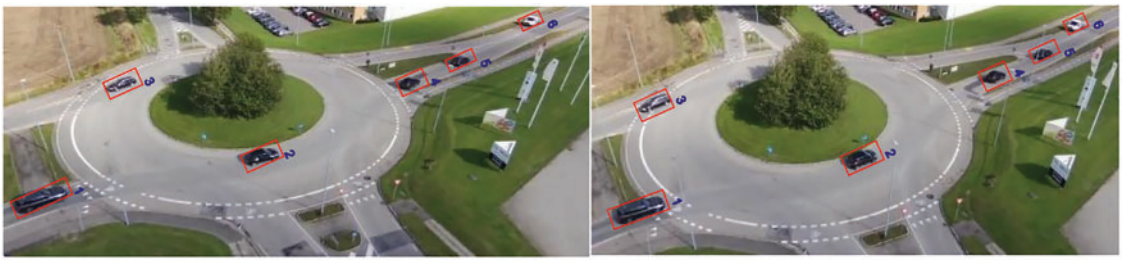


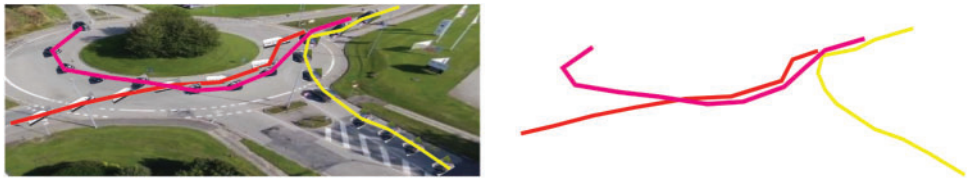
Figure 7: Vehicle tracking using particle filter across image frames

3.8 Trajectories and Density Estimation

Finally, to analyze the vehicle's tracking and paths, trajectories for each corresponding vehicle were drawn by recording the possible location of every vehicle as obtained by the particle filter across the image frames using Eq. (23). The result of estimated trajectories can be visualized in Fig. 8.

$$T_i = \left[\frac{(x_1^i + x_2^i)}{2}, \frac{(y_1^i + y_2^i)}{2} \right] \tag{23}$$

where T_i represents the estimated trajectory of i^{th} vehicle, x_1^i, x_2^i, y_1^i, y_2^i represents the coordinates of each vehicle's location represented by a rectangular bounding box.



(a) (b)

Figure 8: Estimated trajectories of vehicles being tracked (a) trajectories of each vehicle plotted using the centroid points of bounding boxes (b) final output

Also, a record of detected vehicle count was maintained in each frame to estimate the density of traffic on the road as seen in Fig. 9.

Figure 9: Density estimation by using vehicle count displayed at the left corner of each image

4 Performance Evaluation

The dataset utilized for the vehicle detection and tracking system is briefly discussed in this section, along with the findings of three distinct experiments that were used to assess the proposed system and its evaluation against several current state-of-the-art traffic monitoring models [29].

4.1 Dataset Description

We used the following publicly available datasets to develop and test our proposed model.

4.1.1 VAID Dataset

Lin et al. presented the Vehicle Aerial Imaging from Drone (VAID) dataset in 2020 for smart traffic monitoring using vehicle detection and classification. It comprised 6000 images taken from a drone with a final image resolution of 1137×640 after downsizing. All the images are in .jpg format captured at 23.9 frames per second. For reliable imagery acquisition, the drone was positioned between 90 and 95 m above the ground.

4.1.2 AU-AIR Dataset

The 32,823 extracted frames from 8 video segments of more than 2 h make up the AU-AIR dataset [30]. It contains 32,823 labeled frames with 132,034 object instances in total. The images are acquired at the rate of 30 frames per second with a resolution of 1920×1080. Among the multi-modal sensor data in AU-AIR are the altitude, position, time, and velocity. The traffic videos were captured at P.O. Pedersensvej and Skejby Nordlandsvej (Denmark).

4.1.3 UAVDT Dataset

The 100 video sequences which contain 80,000 image frames make up the Unmanned Aerial Vehicle Benchmark Object Detection and Tracking (UAVDT) dataset [31] were selected from more than 10 h of footage taken with an Unmanned Aerial Vehicle (UAV) platform in various urban environments. All the images are in .jpg format with a resolution of 1080×540 pixels acquired at the rate of 30 frames per second. These scenarios include arterial streets, squares, toll booths, motorways, T-junctions, and crossings.

4.2 Experimental Settings and Results

Python (3.7) has been used to design and test the system on a computer with an Intel Core i5 processor running the 64-bit version of Windows 10. The system has 8 Giga Byte Random Access Memory (RAM) and a 2.8 Giga Hertz (GHz) Central Processing Unit (CPU). The performance of the proposed detection and tracking algorithms was evaluated using precision, F1 score, and recall metrics.

4.2.1 Experiment I: Semantic Segmentation Accuracy

The images from each dataset were divided into training and testing samples. 80% samples were used for training and 20% were used for testing purposes. Random Forest classifier was trained as it can increase the accuracy score after fitting different subsamples of the dataset using a variety of decision tree classifiers. The overall accuracy over the training and testing samples was 92% and 77%, respectively.

4.2.2 Experiment II: Precision, Recall, and F1 scores

Table 1 demonstrates the precision, recall, and F1 scores for vehicle detection. True Positive represents the number of vehicles detected successfully. False Positives denote detection other than vehicles, and False Negative represents the number of missed vehicles. The results show that the proposed algorithm can recognize the various vehicles of variable size with high precision.

Table 1: Precision, recall, and F1 scores for vehicle detection

Dataset	Precision	Recall	F1 score
VAID	0.84	0.82	0.83
AU-AIR	0.87	0.86	0.86
UAVDT	0.73	0.78	0.75
Mean precision = 0.81 mean recall = 0.82 mean F1 score = 0.81			

For tracking True Positive represents the number of vehicles successfully tracked, False Positive represents the number of false track of vehicles in more than three frames, and False Negative denotes the number of vehicles not tracked. Table 2 represents the precision, recall, and F1 scores for the proposed tracking algorithm.

Table 2: Precision, recall, and F1 scores for vehicle tracking

Dataset	Precision	Recall	F1 score
VAID	0.88	0.88	0.88
AU-AIR	0.89	0.90	0.89
UAVDT	0.77	0.79	0.78
Mean precision = 0.85 mean recall = 0.86 mean F1 score = 0.85			

4.2.3 Experiment III: ID Assignment and ID Recovery

This section discusses the result of ID assignment and ID recovery to track multiple objects (vehicles) across the different image frames. For this, we used True ID Rate (TIDR) to assess the vehicle ID assignment module and True Recovery Rate (TRR) which represents the number of IDs recovered successfully to assess the recoverability module. For feature matching, if the number of feature matches exceeds 5, then a match was found and the corresponding ID was recovered. In the other case, a new ID was assigned. Table 3 shows the result for the performance evaluation metrics.

Table 3: Measurement of the ID recovery algorithm

Dataset	TIDR (%)	TRR (%)
VAID	40.22	41.32
AU-AIR	43.69	45.36
UAVDT	31.54	39.21

4.2.4 Experiment IV: Comparison with Other Systems

We evaluated our proposed system in comparison with other state-of-the-art methods including deep learning techniques available in the literature. Table 4 demonstrates the comparison of different detection models over the AU-AIR, VAID, and UAVDT datasets. It can be seen that our model outperformed all other techniques in terms of precision. Table 5 presents the advantages and disadvantages of the proposed and existing models.

Table 4: Comparison of the proposed detection model with the state-of-the-art techniques

Datasets	Model	AP
AU-AIR	YOLOv4 [32]	0.81
AU-AIR	YOLOv3 [32]	0.69
AU-AIR	**Our method**	**0.87**
VAID	YOLOv4 [33]	0.83
VAID	Faster R-CNN [33]	0.82
VAID	**Our method**	**0.84**
UAVDT	RetinaNet [34]	33.95
UAVDT	FPN [34]	37.81
UAVDT	NDFT [34]	52.03
UAVDT	D2Det [34]	56.92
UAVDT	**Our method**	**0.73**

Table 5: Advantages and disadvantages of the existing and proposed vehicle detection techniques

Methods	Advantages	Disadvantages
YOLO [32]	YOLO is an efficient detector that can detect objects with high accuracy.	Without eliminating irrelevant areas from the images, YOLO-based detection is computationally expensive.
Faster R-CNN [32]	It is a single, end-to-end solution for the vehicle detection task.	The model does not produce reasonable results if the viewing angle to the ground enlarges.
RetinaNet [33]	RetinaNet is capable of detecting small-sized objects which is important for object detection in aerial images.	It produces very dense candidate frames when increases the computational time as well as increases the number of false positives.
FPN [33]	It is simple and easy to implement.	Object localization is not accurate
NDFT [33]	It can learn domain-robust features thus producing a reasonable precision rate.	As the aerial image contains a large irrelevant area which is processed unnecessarily.

(Continued)

Table 5 (continued)

Methods	Advantages	Disadvantages
D2Det [33]	It produces accurate object localization by exploiting the discriminative features.	The time complexity of the model is high.
Our method	Our proposed method implements a segmentation technique to lower the computational complexity of the model by eliminating irrelevant areas.	The model requires explicit training and testing for segmenting images which limits its applicability.

Table 6 presents a comparison of the tracking algorithm with the other tracking techniques. Whereas, Table 7 compares the advantages and disadvantages of the existing and proposed model.

Table 6: Comparison of the proposed tracking model with the state-of-the-art techniques

Datasets	Model	Precision
AU-AIR	SIFT features [35]	0.75
AU-AIR	Shape-based matching [25]	0.51
AU-AIR	**Our method**	**0.89**
VAID	SIFT features [35]	0.73
VAID	Shape-based matching [25]	0.48
VAID	**Our method**	**0.88**
UAVDT	SIFT features [35]	0.58
UAVDT	Shape-based matching [25]	0.32
UAVDT	**Our method**	**0.77**

Table 7: Advantages and disadvantages of the existing and proposed vehicle tracking techniques

Methods	Advantages	Disadvantages
SIFT [34]	SIFT features are invariant to scale and rotation which makes them feasible for object-tracking tasks.	If the objects appear too small then SIFT alone cannot produce good results.
Shape-based matching [24]	It is simple and easy to implement.	It requires explicit vehicle modeling for each detection which limits the tracking capabilities, especially in dense traffic conditions.
Our method	The particle filter combined with ID assignment is an efficient and robust method to track multiple vehicles across the image frames.	The tracking highly depends on the detection algorithm output.

5 Discussion

The proposed system is an effective solution for intelligent traffic monitoring using aerial images. Object recognition in high-resolution aerial images is a very challenging task. Therefore, we proposed a mechanism based on multi-label semantic segmentation and particle filter-based tracking to achieve efficient results. However, the proposed method has some drawbacks. First of all, the system is only tested on RGB images captured during day time. The approach can be further validated by evaluating image and video datasets captured at night or in low-light situations since many researchers have had success with these datasets. Moreover, our segmentation and detection algorithm faces difficulty under partial or full occlusions, roads covered with trees, or similar objects. Fig. 10 shows a nighttime image from the UAVDT dataset.

Figure 10: Drawbacks of the vehicle detection algorithm. (a) Different illumination conditions at night time (b) vehicle not detected due to background cluttering (c) vehicle not detected due to occlusion

6 Conclusion and Future Works

In this paper, an effective system for vehicle detection and tracking under various road circumstances is proposed. The RGB images are first all segmented into five classes and then the images are subjected to mean-shift clustering for noise removal and to smoothen the output. After that, vehicle pixels are extracted and a blob detection technique was applied to detect each vehicle. Each vehicle was verified using ground truth labeling. To track multiple vehicles, each of them was assigned an ID based on ORB features. The proposed model produces significant results on all three datasets which proves the effectiveness of our methodology.

To increase performance in the future, the authors want to test new and improved classifiers on more complicated and varied datasets. Moreover, to improve the performance of the traffic monitoring system we aim to use deep learning methods.

Acknowledgement: The authors are thankful to Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) Support Program (IITP-2023-2018-0-01426) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). The funding of this work was provided by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Asifa Mehmood Qureshi, Jeongmin Park; data collection: Nouf Abdullah Almujally; analysis and interpretation of results: A. Qureshi, Saud S.

Alotaibi and Mohammed Hamad Alatiyyah; draft manuscript preparation: Asifa Mehmood Qureshi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All publicly available datasets are used in the study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Rad and M. Jamzad, "Real time classification and tracking of multiple vehicles in highways," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1597–1607, 2005.

[2] S. K. Weng, C. M. Kuo and S. K. Tu, "Video object tracking using adaptive Kalman filter," *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190–1208, 2006.

[3] M. Alarfaj, M. Waheed, Y. Ghadi, S. Tamara, A. Suliman *et al.,* "An intelligent framework for recognizing social human-object interactions contextual scene understanding," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 1207–1223, 2022.

[4] N. Khalid, Y. Y. Ghadi, M. Gochoo, A. Jalal and K. Kim, "Semantic recognition of human-object interactions via Gaussian-based elliptical modeling and pixel-level labeling," *IEEE Access*, vol. 9, pp. 111249–111266, 2021.

[5] H. Zhang and K. Wu, "A vehicle detection algorithm based on three-frame differencing and background subtraction," in *Proc. of 2012 Fifth Int. Symp. on Computational Intelligence and Design*, Hangzhou, China, vol. 1, pp. 148–151, 2012.

[6] Z. Yi, T. Chang, S. Li, R. Liu, J. Zhang *et al.,* "Scene-aware deep networks for semantic segmentation of images," *IEEE Access*, vol. 7, pp. 69184–69193, 2019.

[7] J. Zhang, Q. Zhang and C. Shi, "An unmanned aerial vehicle detection algorithm based on semantic segmentation and visual attention mechanism," in *Proc. of the 2018 2nd Int. Conf. on Computer Science and Artificial Intelligence*, Shenzhen, China, pp. 309–313, 2018.

[8] R. Makrigiorgis, N. Hadjittoouli, C. Kyrkou and T. Theocharides, "AirCamRTM: Enhancing vehicle detection for efficient aerial camera-based road traffic monitoring," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, Waikoloa, HI, USA, pp. 3431–3440, 2022.

[9] A. Gomaa, M. M. Abdelwahab and M. Abo-Zahhad, "Efficient vehicle detection and tracking strategy in aerial videos by employing morphological operations and feature points motion analysis," *Multimedia Tools and Applications*, vol. 79, no. 35, pp. 26023–26043, 2020.

[10] Y. Wu, H. Guo, C. Chakraborty, M. Khosravi, S. Berretti *et al.,* "Edge computing driven low-light image dynamic enhancement for object detection," *IEEE Transactions on Network Science and Engineering*, vol. 10, pp. 1, 2022.

[11] M. Teutsch, W. Krüger and J. Beyerer, "Moving object detection in top-view aerial videos improved by image stacking," *Optical Engineering*, vol. 56, no. 8, pp. 083102–083102, 2017.

[12] X. Kong, Q. Chen, G. Gu, K. Ren, W. Qian *et al.,* "Particle filter-based vehicle tracking via HOG features after image stabilisation in intelligent drive system," *IET Intelligent Transport Systems*, vol. 13, no. 6, pp. 942–949, 2019.

[13] H. Gupta and O. P. Verma, "Monitoring and surveillance of urban road traffic using low altitude drone images: A deep learning approach," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19683–19703, 2022.

[14] M. Ozturk and E. Cavus, "Vehicle detection in aerial imaginary using a miniature CNN architecture," in *Proc. of 2021 Int. Conf. on Innovations in Intelligent Systems and Applications (INISTA)*, Kocaeli, Turkey, pp. 1–6, 2021.

[15] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan *et al.,* "Deep learning approach for car detection in UAV imagery," *Remote Sensing*, vol. 9, no. 4, pp. 312, 2017.

[16] H. C. Baykara, E. Biyik, G. Gul, D. Onural and A. S. Ozturk, "Real-time detection, tracking and classification of multiple moving objects in UAV videos," in *Proc. of 2017 IEEE 29th Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, Boston, MA, USA, pp. 945–950, 2018.
[17] A. A. Rafique, M. Gochoo, A. Jalal and K. Kim, "Maximum entropy scaled super pixels segmentation for multi-object detection and scene recognition via deep belief network," *Multimedia Tools and Applications*, vol. 82, pp. 1–30, 2022.
[18] A. A. Rafique, A. Jalal and K. Kim, "Automated sustainable multi-object segmentation and recognition via modified sampling consensus and kernel sliding perceptron," *Symmetry*, vol. 12, no. 11, pp. 1928, 2020.
[19] A. Jalal, A. Ahmed, A. A. Rafique and K. Kim, "Scene semantic recognition based on modified fuzzy C-mean and maximum entropy using object-to-object relations," *IEEE Access*, vol. 9, pp. 27758–27772, 2021.
[20] V. Grau, A. U. J. Mewes, M. Alcañiz, R. Kikinis and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, 2004.
[21] X. Zeng, I. Chen and P. Liu, "Improve semantic segmentation of remote sensing images with K-mean pixel clustering: A semantic segmentation post-processing method based on k-means clustering," in *Proc. of 2021 IEEE Int. Conf. on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*, Barcelona, Spain, pp. 231–235, 2021.
[22] A. A. Rafique, A. Al-Rasheed, A. Ksibi, M. Ayadi, A. Jalal *et al.,* "Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images," *IEEE Access*, vol. 11, pp. 2993–3007, 2023.
[23] G. N. Chaple, R. D. Daruwala and M. S. Gofane, "Comparisions of robert, prewitt, sobel operator based edge detection methods for real time uses on FPGA," in *Proc. of Int. Conf. on Technologies for Sustainable Development (ICTSD)*, Mumbai, India, pp. 1–4, 2015.
[24] K. L. Wu and M. S. Yang, "Mean shift-based clustering," *Pattern Recognition*, vol. 40, no. 11, pp. 3035–3052, 2007.
[25] J. Leitloff, D. Rosenbaum, F. Kurz, O. Meynberg and P. Reinartz, "An operational system for estimating road traffic information from aerial images," *Remote Sensing*, vol. 6, no. 11, pp. 11315–11341, 2014.
[26] U. Sara, M. Akter, M. S. Uddin, U. Sara, M. Akter *et al.,* "Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.
[27] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. of IEEE Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 2564–2571, 2011.
[28] H. Chu, K. Wang and X. Xing, "Target tracking via particle filter and convolutional network," *Journal of Electrical and Computer Engineering*, vol. 2018, pp. 1–9, 2018.
[29] M. Asifa and J. Ahmad, "Vehicle detection and tracking using Kalman filter over Aerial images," in *Proc. of Int. Conf. on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, pp. 1–6, 2023.
[30] I. Bozcan and E. Kayacan, "AU-AIR: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance," in *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Paris, France, pp. 8504–8510, 2020.
[31] H. Yu, G. Li, W. Zhang, Q. Huang, D. Du *et al.,* "The unmanned aerial vehicle benchmark: Object detection, tracking and baseline," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1141–1159, 2020.
[32] H. Xu, Y. Cao, Q. Lu and Q. Yang, "Performance comparison of small object detection algorithms of UAV based aerial images," in *Proc. of 9th Int. Symp. on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Hongkong, China, pp. 16–19, 2020.
[33] H. Y. Lin, K. C. Tu and C. Y. Li, "VAID: An aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212209–212219, 2020.

[34] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang *et al.,* "D2DET: Towards high quality object detection and instance segmentation," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 11482–11491, 2020.

[35] K. Mu, F. Hui and X. Zhao, "Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching," *Journal of Information Processing Systems*, vol. 12, no. 2, pp. 183–195, 2016.