A Systematic Review of Drone Based Road Traffic Monitoring System

IGOR BISIO (Senior Member, IEEE), CHIARA GARIBOTTO (Member, IEEE), HALAR HALEEM (Member, IEEE), FABIO LAVAGETTO (Member, IEEE), AND ANDREA SCIARRONE (Senior Member, IEEE)
Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture (DITEN), University of Genoa, 16145 Genoa, Italy
Corresponding author: Andrea Sciarrone (andrea.sciarrone@unige.it)

ABSTRACT Drone deployment has become crucial in a variety of applications, including solutions to traffic issues in metropolitan areas and highways. On the other hand, data collected via drones suffers from several problems, including a wide range of object scales, angle variations, truncation, and occlusion. To process and manipulate visual data from the drones, a variety of image processing algorithms have been employed, each with a distinct aim. Additionally, recent breakthroughs in the field of Artificial Intelligence, particularly deep learning, have attracted broad interest and are being applied to many domains in the framework of smart cities, including road traffic monitoring. The purpose of this study is to conduct a systematic review of drone-based traffic monitoring systems from a deep learning perspective. This work focuses on vehicle detection, tracking, and counting, since they are fundamental building blocks towards founding solutions for traffic congestion, flow rate and vehicle speed estimation. Additionally, drone-based datasets are examined, which face issues and problems caused by the diversity of features inherent of drone devices. The review analysis presented in this work summarizes the literature solutions provided and deployed so far and discusses future research trends in establishing a comprehensive traffic monitoring system in support of the development of smart cities.

INDEX TERMS Deep learning, drone, smart city, traffic monitoring, UAV.

I. INTRODUCTION
Transportation, a critical use case for smart cities, spans a variety of areas, including road safety, highway infrastructure management, and traffic monitoring [1]. Many tasks have been defined in the framework of traffic monitoring, including vehicle identification, counting, tracking, road accidents and congestion detection, and vehicle speed estimation [2], [3], [4], [5]. Unmanned Aerial Vehicles (UAVs) are being used in a plethora of applications in the modern era, most noticeably in monitoring and surveillance tasks related to the development of smart cities, due to the significant benefits they offer, including smart altitude and location management, dynamic coverage range, real-time data collection and processing, hurdle avoidance, and the surveillance of small/large, static/moving objects [3], [6], [7], [8]. Various technologies, including the Internet of Things (IoT), Artificial Intelligence (AI), 5G, Edge Computing, and Cloud Computing, are being used to build up the whole concept of smart city. UAVs, in conjunction with AI, are becoming a market leader in delivering solutions for various traffic monitoring tasks, such as traffic congestion, in order to transform existing cities’ structures into smart facilities and services [9]. Notwithstanding the benefits of using UAVs, their peculiar usage has raised public privacy issues too [10]. Moreover, utilization of such devices impose difficulties and challenges that must be addressed. For instance, video obtained from lower altitudes in urban areas raises privacy concerns. Performing the same activity from a higher altitude, on the other hand, would require high resolution data to capture relevant information. On the one hand, recording high-resolution UAV video gathers sufficient ground data on individual vehicles for traffic monitoring activities, such as their trajectory, lane change information, and vehicle interaction [3]. On the
other hand, these high-resolution videos inherently complicate detection and tracking tasks [6]. Other important aspects such as the appearance of small objects and variations in their view points, occlusion, truncation, and illumination changes jointly contribute to the difficulty of processing the videos to extract useful information [11]. Additionally, dynamic moving backgrounds and complex environments complicate the estimation of ground vehicle speeds [12].

Recent AI advances, particularly Deep Learning (DL), are being used in smart city initiatives like traffic monitoring. Traffic monitoring tasks such as traffic congestion, flow rate analysis, and so on are heavily reliant on core task like vehicle detection, counting, and tracking. The development of smart traffic monitoring systems urges to explore and identify the various methods and concepts based on DL approaches implemented so far to design these fundamental tasks. To systematically analyze and analytically examine a smart traffic monitoring system, it is also worth investigating other factors such as datasets, pre-processing techniques, evaluating metrics, and the nature of the development system (for instance, remote or on-board).

The purpose of this paper is to present an overview of the state-of-the-art DL solutions in order to devise an efficient road traffic monitoring system. In this connection, two specific objectives have been defined:

i. To investigate the drone-based road traffic monitoring systems that have been proposed and developed so far using DL approaches.
ii. To explore the architectures, pre-processing techniques and datasets, that have been primarily used to implement drone-based traffic monitoring systems in the framework of vehicle detection, tracking and counting.

As an outcome of this study, a survey of recent developments in the design of DL-based traffic monitoring systems has been provided. In particular, information regarding datasets and their properties, pre-processing strategies, frameworks, and methodologies that have been employed for various tasks such as vehicle detection, tracking and counting tasks, is discussed. In the methodology part, the Research Questions (RQs) are formulated, which serve as the focal point of the current study being researched. Their purpose is to investigate the prior work in the field under consideration, and are developed in accordance with the objectives of the review study. The overall workflow of the research study carried out is depicted in Fig. 1. This study is intended as a baseline for future trends and developments in the perspective of traffic monitoring systems based on the use of drones and DL techniques.

This survey work is structured as follows: background information and methodology are described in Sections II and III, respectively. Sections IV to VI, cover the findings presented for the considered RQs, where Section VI also highlights potential future research trends. Finally, in the last Section conclusions are drawn.

II. BACKGROUND AND MOTIVATION

According to a study from the United Nations, the global population is predicted to grow by two billion people by 2050 [13]. Consequently, issues concerning transportation, both related to infrastructure and traffic monitoring, are rising as a result of high population growth, thus driving the rapid development of smart cities. Unlike traditional cameras deployed for traffic surveillance, drone-based vision enables a diverse and smart system to be designed. A general framework for AI-based traffic monitoring using UAVs can be outlined as follows: acquisition of images/videos, data processing using AI/DL techniques, output to the assigned users, and management centers [9]. In this connection, a detailed guide on implementing a UAV-based traffic monitoring system is given in [14]. UAV-based traffic monitoring tasks can be carried either on-board or via cloud-based video processing. Fig. 2 illustrates a high-level schematic view of the processing and analysis of cloud-based aerial images/videos.

As discussed above, implementing drone-based traffic monitoring tasks in urban areas and on highways is quite...
challenging due to the multitude of challenges in the acquired videos and images. This ultimately makes it more difficult to attain the highest level of accuracy for various traffic monitoring tasks (vehicle identification, tracking, counting, and speed estimation, for example). Broadly, vehicle object detection in images/videos is accomplished via two approaches: separation of foreground (moving) and background (static) objects [15] and features (such as color, texture, shape, etc., [16]) extraction based detection. In the former approach, techniques such as background subtraction [17], [18] are employed, while in the latter approach, different object features including complex ones such as Speed Up Robust Feature (SURF) [19], Scale Invariant Feature Transformation (SIFT) [20], Haar-like features [21], and Histogram of oriented Gradients (HoG) [22], are utilized. In comparison to these two approaches, recent DL-based solutions such as [2], [3], [23], [24] have been widely adopted due to their robustness and improved vehicle detection performance in images and videos.

Several object detection frameworks have been reported with the advent of DL-based frameworks, including one-stage detectors [25], [26], as well as two-stage detectors [27], [28]. Mostly, DL-based frameworks use pre-trained models to improve and refine the feature extraction, such as in [29]. In [30], a review of some of the main state-of-the-art object detection frameworks is provided along with some experimental analysis. In contrast to static images, object detection in videos requires processing of each frame. Additionally, the following concepts must be taken into account: the spatial and temporal correlation of frames to overcome feature extraction redundancies, and the effect of motion blur, occlusion, and posture changes, which can contribute to the low quality of some individual frames in a video sequence [31]. Indeed, overlooking these factors results in a decrease in the object detection performance in videos. The difference between static and video object detection frameworks lies in the incorporation of temporal information. Furthermore, although video object detection and Multi-Object Tracking (MOT) both use UAV video data, the way temporal information is employed differs in the two cases. The former aims at improving the detection rate of the current frame by exploiting context information from previous frames, while the latter aims at forecasting the trajectory of objects in future frames. The difference between these two approaches is shown in Fig. 3.

Furthermore, in UAV-based videos the movements of the camera within a scene, usually referred to as ego motion, complicate vehicle detection and other traffic monitoring tasks significantly. Prior to the advent of DL-based frameworks, this issue was addressed in two distinct ways: image registration [32], [33], [34], [35] and optical flow [33], [34], [36]. In image registration solutions, moving background is turned into a fixed one, hence facilitating the background subtraction task [17] to overcome the ego motion issue. Alternatively, optical flow is also used in conjunction with image registration, although according to [37], optical flow and background subtraction with image registration are ineffective in detecting vehicles in dense traffic situations. However, optical flow combined with supervised learning enables the detection of vehicles in a dense traffic environment using a UAV video with ego motion.

III. METHODOLOGY

Keeping in view the need and importance of drone-based traffic monitoring system using DL, a systematic study has been carried out on three domains characterizing traffic monitoring systems: vehicle detection, tracking and counting. Various frameworks, methodologies, and DL-based solutions have been adopted and validated so far to design such systems. To conduct the review, the following RQs have been devised in relation to the identified objectives:

RQ1: Which drone-based aerial datasets have been used so far for vehicle monitoring systems and which characteristics have been considered of these datasets? Also, which common evaluating metrics have been used in the literature?

RQ2: Exploration of drone based road traffic monitoring systems that have been implemented in the perspective of DL. More specifically:

a) Which frameworks and techniques have been used for the design of such systems in the context of vehicle detection, tracking and counting?

b) Which pre-processing steps/techniques have been used and are effective?

RQ3: a) What is their effectiveness, significance & generalization based upon their performance evaluation?

b) Based upon the findings, what could be the possible challenges and future road map towards the
TABLE 2. Summary of screened articles with respect to the specific task of traffic monitoring system.

| Parameter       | Detection | Counting | Tracking |
|-----------------|-----------|----------|----------|
|                 | Section V.A, Section V.A2, Section V-B | Section V-D | Section V-E |
| Journal         | 18        | 4        | 9        |
| Conference      | 13        | 3        | 4        |
| Task wise total | 31        | 7        | 13       |
| Total           | Detection + Counting + Tracking = 32 |

realization of a traffic monitoring system capable of performing detection, tracking and counting tasks more efficiently? Following the completion of the research objectives and formulation of the RQs, a systematic approach is used to collect all relevant research studies. *IEEE Xplore, Scopus,* and *Google Scholar* electronic databases have been used to identify pertinent material. The *Journal* category received the most attention, followed by *Conference* papers. Moreover, other documents such as *Theses, Book Chapters,* and *Technical Reports* were also screened and reviewed accordingly.

To the purpose of the present study, a two-step approach is used: first, publications are screened based on their title and abstract, and then the most relevant papers are shortlisted after full-text reading. Additionally, references to relevant articles are scanned for inclusion of supporting materials and evidences. The publication year range was considered from 2015 to 2021 due to the remarkable increase in DL-based solutions. The eligibility criteria and considered reference points (keywords, electronic databases, language, and publication era) in carrying out the review method and its findings (number of screened and included articles) are detailed in Table 1. In addition, Table 2 shows the segregation of screened articles by article type and number of articles in each task of the traffic monitoring system. Since some of the articles tackled more than one task and the number of task-specific articles is greater, the last row represents the union of the number of articles that adds up to the total number of shortlisted articles.

IV. AERIAL DATASETS AND EVALUATING METRICS (RQ1)

A. DATASETS

Designing accurate machine learning and DL models capable of detecting, localizing and identifying multiple objects in an image is still a core challenge in the field of computer vision. The diversity of drones has enhanced their use in industrial, commercial, and security applications. Visual monitoring, surveillance and analysis are limited with fixed-angle cameras. However, drones with moving cameras, as an alternative, provide a large field of view, high mobility, and ease of deployment. On the other hand, moving cameras have made DL-based object detection and tracking challenges increasingly complex. In the following we highlight the main challenges related to such tasks:

- The speed of UAVs, camera rotation, and moving objects all affect overall performance and must be properly examined.
- Since UAVs fly at high altitudes, target size is reduced, making it difficult to recognize and track small objects. Moreover, varying altitudes and shooting angles produce arbitrary object shapes and orientations that must be tackled.
- Real-time concerns are also important in practical deployment, since systems can be either designed on-board or managed remotely by receiving video streams from UAVs.

To address these challenges, the availability of appropriate datasets is a fundamental step towards finding suitable algorithmic solutions. In this connection, Convolutional Neural Networks (CNNs) are typically employed for image processing, and they are data hungry [48]. In recent years, researchers have produced various drone-based aerial datasets to address data availability issues for different applications: among others, UAV-Bottle Dataset (UAV-BD) dataset [49] to detect the waste bottles, UAV Mosaicking and Change Detection (UMCD) dataset [50] to improve the algorithms for both change detection and mosaicking on low-altitude aerial video sequences, Okutama-Action [51] for human action detection are worth to mention. We analyzed the main drone-based datasets related to traffic monitoring applications, which we summarize in the Table 3. They contain images and video sequences acquired under various conditions. The dataset prepared in [46] contains 110k frames to the purpose of object tracking from an aerial view. This dataset has been acquired at low UAV altitude, with moving camera, and different shooting angles, due to UAV motion. These features make this dataset suitable for drone monitoring, although it lacks information on dynamic weather and higher altitudes. Moreover, 100 aerial sequences from a drone were acquired to the aim of object detection, Single Object Tracking (SOT) and MOT in [45]. In this work, a wide range of urban settings, including crossings and T-junctions, were considered to obtain the image data, making it appropriate for traffic monitoring and surveillance systems.

Another relevant dataset is Visdrone [39], [40], the largest drone-based aerial dataset, which contains four tasks: object detection in images, videos, SOT, and MOT. It includes 263 video sequences and more than 10000 static images, all of which were collected using various drones in 14 different cities in diverse weather and lighting conditions and at various altitudes. Likewise, UAV Detection and Tracking (UAVDT) dataset [45] provides information for video object detection, SOT, and MOT tasks with vehicles as the target objects. Furthermore, in [42], an aerial dataset of a road scene is produced using two different points of view: a static camera and a drone; each video consist of 3100 frames, and the two acquisitions are used to investigate the differences between the two viewpoints. Four object categories have been labeled at the pixel level in this dataset. Similarly
TABLE 3. Summary of the main drone-based aerial imagery datasets.

| Task                  | Dataset          | Object classes | Images/Frame | Resolution | Altitude | Occlusion | Camera view(s) | Year |
|-----------------------|------------------|----------------|--------------|------------|----------|-----------|----------------|------|
| Object detection (images) | CARPK [38]      | 1              | 1.5k         | 1280 × 720 | 40m      | –         | Single         | 2017 |
|                        | Valdrone-DST [39], [40] | 10             | 10.209k      | 2000 × 1200 | –        | –         | Multiple        | 2018, 2019 |
|                        | DAC-EDC [41]     | 95             | 150k         | 640 × 360  | Low      | –         | Multiple        | 2018 |
|                        | MTIR [42]        | 4              | 3.1k         | 1920 × 1080 | –        | –         | Single         | 2020 |
|                        | USVD [43]        | 1              | 5.8k         | 960 × 540 × 5280 × 2970 | 10-150m  | ✓         | Multiple        | 2020 |
|                        | MOHR [44]        | 5              | 10.81k       | 5482 × 3076, 7360 × 4972, 1688 × 5792 | 200m, 300m & 400m | ✓         | Multiple        | 2021 |
| Object detection (videos) | Valdrone-VID [45], [46] | 10             | 40k          | 3640 × 2160 | –        | –         | Multiple        | 2018, 2019 |
|                        | UAV2012 [48]     | 3              | 80k          | 1080 × 640  | 10-70m   | –         | Multiple        | 2018 |
|                        | UAV21 [49]       | 3              | 80k          | 1080 × 540  | 10-70m   | –         | Multiple        | 2018 |
|                        | Valdrone-SOT [48], [50] | 3              | 139k         |           |          |           | Multiple        | 2018, 2019 |
|                        | UDAV23 [46]      | –              | 110k         | 720 × 720   | 5-25m    | –         | Multiple        | 2014 |
|                        | UAV2012 [48]     | –              | 80k          | 1080 × 540  | 10-70m   | –         | Multiple        | 2018 |
|                        | Standard drone [47] | 6              | 929-499k     | 1400 × 1904 | 80m      | ✓         | Single         | 2016 |
|                        | UAV2012 [48]     | 3              | 80k          | 1080 × 640  | 10-70m   | –         | Multiple        | 2018 |
|                        | Valdrone-MOT [39], [40] | 10             | 40k          | 3640 × 2160 | –        | –         | Multiple        | 2018-2019 |
| Object counting      | CARPK [38]       | 1              | 1.5k         | 1280 × 720  | 40m      | –         | Single         | 2017 |

B. EVALUATION METRICS

1) FOR DETECTION TASK

Two main categories of evaluating metrics have been used in the literature to assess the performance of object detection systems. The most widely employed is the accuracy in terms of precision, recall and Average Precision (AP), followed by the detection speed defined as the number of frames the detector is capable to process per second. In general, binary predictions are usually classified with respect to the Ground Truth (GT) as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). However, in the case of object detection, a large number of Bounding Boxes (BBoxes) are generated and the majority of them are not required to be predicted. In this situation, a large number of empty BBoxes are correctly detected as non-object, hence the prediction parameter TN is not used since it is not relevant to the object detection task. As a consequence, metrics such as True Positive Rate (TPR), False Positive Rate (FPR), and Receiver Operating Characteristic (ROC) curves, that are based on the TN prediction parameter are commonly ignored during evaluation [52].

The output of a particular detector is predicted as \( (b_i, s_i, I_i, t)_{i=1}^n \) for a given input image, where \( b_i, s_i \) and \( I_i \) are the predicted BBoxes, scores and labels respectively for the \( i \)th object, and \( n \) is the number of detections. Before any detector could predict, threshold values such as confidence threshold \( C_t \) and IoU threshold \( I_i \) are assigned. If the predicted label \( l_i \) matches the GT label, the predicted score \( s_i \) is greater than \( C_t \), and the IoU value is greater than the \( I_i \), the predicted output \( (b_i, s_i, I_i) \) is classified as TP, otherwise it is considered FP. The evaluating metrics, precision and recall, of the object detection algorithm are defined as follows:

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{1}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{2}
\]

Precision measures the true positive predictions among all positive predictions, while recall refers to the same concept applied to all predictions, both positive and negative. Other metrics, such as quality, completeness, and correctness, are also used to evaluate the performance of object detection frameworks, for instance in [53]. Completeness is similar to recall, while correctness is equivalent to precision. Compared to completeness and correctness, the significance of the quality metric remains high because it incorporates both of them and calculated as follows:

\[
\text{Quality} = \frac{TP}{TP + FP + FN} \tag{3}
\]

Since precision and recall metrics do not include TN prediction, the Precision-Recall (PR)-curve is also employed as an evaluating metric to examine the performance of the detector, particularly for the unbalanced dataset, as given in [45]. For a given confidence threshold \( C_t \) in the detector, an optimal RP-curve should have high precision and recall values, which means a large Area Under the Curve (AUC). However, in real experiments, there is always a zig-zag pattern for PR-curve due to noise, which makes true measurement of AUC for precision and recall values quite difficult. The zig-zag pattern is removed before estimating the AUC, and this is done by computing the AP using either 11-point interpolation or all-point interpolation. The former consists in interpolating the precision value for 11 recall points e.g., \( (0, 0.1, \ldots, 1) \). This method was initially proposed in Pascal-VOC competition [54] and later changed to all-point interpolation. Adapting the all-point interpolation approach increases the...
chance of closely estimating the AP value. In such case, the resultant AP value is usually slightly greater than that obtained with the 11-point interpolation approach. Increasing the IoU threshold, on the other hand, decreases the AP value, and vice versa. This is because as the IoU threshold grows, the TP predictions decrease, making the model more robust. We refer the reader to [52], [54], [55], [56], [57] for further details about evaluating metrics related to object detectors. In the case of a multi-class problem, predictions are made separately for each category, and the mean AP (mAP) metric is used to assess the overall performance of object detectors. This value is obtained by averaging the AP values of m object classes, as follows:

\[
mAP = \frac{1}{m} \sum_{j=1}^{m} AP_j
\]

In the explored work for traffic monitoring systems, the AP metric has been mainly used to assess vehicle detection in aerial images and videos. Additionally, metrics such as precision, accuracy, and F1 scores are employed. To maintain the uniformity of these metrics, attempts are made to record the evaluation figures of the researched literature under comparable metrics.

2) FOR TRACKING TASK

The performance of state-of-the-art object trackers has been evaluated using the following metrics [58], [59], [60]:

i) **Recall**: It indicates ratio of correctly matched objects among all GT objects.

ii) **Precision**: It measures the ratio of correctly matched objects among all output objects.

iii) **False Alarms**: The number of FAs/frame.

iv) **FP**: The number of false positives or false detections.

v) **FN**: The number of FN/frame or missed detections.

vi) **GT**: The number of GT trajectories.

vii) **Mostly Tracked (MT)**: The number of target trajectories covered by tracker for more than or at least 80% of the trajectory length.

viii) **Mostly Lost (ML)**: The number of target trajectories covered by tracker for less than or at least 20% of the trajectory length.

ix) **Fragments (Frag)**: The number of times a tracked trajectory is interrupted.

x) **Identity Switches (IDS)**: The number of times an object ID changes to another ID.

In an optimal tracker, precision, recall, MT, and GT metrics should provide high values, whereas FA, FN, FP, ML, Frag, and IDS indicators should be low. In addition, the literature proposes the following two standard MOT evaluation metrics [61].

- **MOT Accuracy (MOTA)**: This depends upon the three error sources, i.e., FP, FN and IDs.

- **MOT Precision (MOTP)**: It reflects the average dissimilarity between the obtained true positives and corresponding GT targets.

MOTA and MOTP values should be higher for an efficient and accurate tracker. Furthermore, the VisDrone2018 competition [62] employed alternate evaluating metrics for the two MOT tracking tasks, defining MOT-a and MOT-b based on the prior availability of object detection data in each video frame. For MOT-a (no prior object detection results), AP-based evaluating metrics were employed [63], while concerning MOT-b, evaluating metrics from [64] were used (with prior detection results). For the performance evaluation of MOT algorithms in Visdrone-2019 and later [40], [65], authors used the metrics from [63], irrespective of the availability of prior detection outcomes. Regarding the evaluation of tracking tasks in traffic monitoring systems, MOTA and MOTP metrics have been primarily employed. In addition to these two, precision and AP have been computed. Differently, some works have reported tracking performance by illustrating motion tracks, such as in [66], while others have only presented qualitative analysis, for instance in [67]. Also, tracking rate, which is the ratio of accurately tracked vehicles to all tracked vehicles, is used in [3].

3) FOR COUNTING TASK

The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics have been employed in the DL-based counting frameworks [2], [38]. Differently, counting tasks employing the counting-by-tracking and lane-based counting approaches usually rely on traditional counting accuracy [37].

V. TRAFFIC MONITORING SYSTEMS: RQ2

This section discusses two main aspects of RQ2. The first part addresses and presents data from explored drone-based traffic monitoring systems in the context of detection, tracking, and counting, while the second part discusses the image pre-processing and augmentation techniques. We choose to present works related to the different aspects separately: vehicle detection first, followed by the vehicle tracking and counting sections. Additionally, on-board traffic monitoring systems are mentioned individually. Based on the conducted research, Fig.4 depicts the taxonomy of traffic monitoring systems for various properties such as processing platform, subsystems/tasks, environment type, issues addressed, and UAV state. Noteworthy is the fact that drone-based traffic monitoring systems have implemented detection, counting, tracking, congestion analysis, flow rate, and speed estimation tasks. This study focuses solely on the detection, tracking, and counting tasks, as indicated by the dashed lines. While implementing subsystems/tasks, the traffic monitoring system exhibits diverse characteristics. For instance, some of the work involved on-board computation of traffic monitoring tasks, while others involved remote processing. Similarly, few studies addressed occlusion problems, whereas others concentrated on real-time processing.

A. TRAFFIC OBJECT DETECTION

1) IMAGE-BASED DETECTION

Table 4 provides a comparison of various vehicle detection systems in drone imagery that utilize DL techniques.
In details, data pertaining to the implemented frameworks and their performance for the specific dataset, characteristics of the dataset, the number of object types, and their outcome are reported therein. If the dataset is small, training a DL model from scratch is not advantageous. In this connection, the concept of transfer learning is widely used. This concept relies upon using a pre-trained model to recognize new targets belonging to selected classes. However, since drone images have aerial views with occlusions, varying sizes and orientations, employing DL models pre-trained on datasets such as Imagenet [56] and MS-COCO [77] datasets might not provide high performance. For this reason, Le et al. [68] proposed a vehicle classification system for drone imagery that combines transfer learning features from the Inception-ResNet-v2 model [78] with hand-made features from HoG [22], Local Binary Patters (LBP) [79] and Bag-of-Visual-Words (BoVM) [80]. The Inception-ResNet-v2 model and BoVM features work well together in recognizing and classifying vehicles in the Vehicle Recognition in Drone Imagery (VRDI) dataset.

Since variable UAV altitudes acquire aerial images of vehicles in different scales, Scale-Specific Prediction-Single Shot MultiBox Detector (SSP-SSD), which is a framework for multi-scale vehicle detection in high resolution (3840 × 2160) UAV images was proposed by Li et al. [69]. Such solution employed ResNet-101 as a DL backbone and added layers to extract features for vehicles of different sizes. The proposed Outlier-Aware Non Maximum Suppression (OA-NMS) also minimized the occurrence of false alarms and missed detections. The proposed approach outperformed many state-of-the-art techniques and two-stage detection frameworks on the custom created dataset. Furthermore, to address the issue of small and dense objects in aerial scenes, Li et al. [2] proposed the Scale Adaptive-Circular Flow (SA-CF) framework. In this framework, a SA strategy was developed to match the different sizes of objects in aerial scenes with the GT BBBoxes by formulating a loss function. A CF is embedded in the feature extractor to maximize the feature extraction information where bottom-up features are combined with the top-down attention features. Nonetheless, this strategy provides scarce performance when compared with SSD [81]. You Only Look Once (YOLO) [80], and YOLOv3 [25] frameworks. Following another approach to account for the features of small size objects, the authors in [70] presented De-convolutional YOLO (DYOLO), in which all layers following conv5_5 and conv6_5 were eliminated from the backbone model, and extra convolutional layers for up-sampling the features to take advantage of context information were added. As a consequence, DYOLO outperformed Faster Region-based CNN (Faster R-CNN) [83], SSD [81], and YOLOv2 [82] frameworks. In addition, to improve vehicle detection performance in high resolution multi-scale aerial images, Li and Li [71] proposed Image Spatial Pyramid Detection Model (ISPD-M) with YOLOv3 framework. For the original and image patch layers of the image spatial pyramid, an integrated decision-making algorithm was formulated to reduce multiple detections of same objects. Results increased over base YOLOv3 [25], Faster R-CNN [83] and SSD [81] frameworks. Moreover, Benjdira et al. [72] used Faster R-CNN [83] and YOLOv3 [25] to detect cars for traffic monitoring purpose.

To analyze the traffic from UAV images, Adaimi et al. [73] designed a Butterfly detector to handle the difficulties of wide range of object scale, viewing angle fluctuation, and occlusion. This is achieved by introducing the butterfly field concept, which describes the spatial information of output characteristics and object scale, while occlusion and viewing angle difficulties are addressed by using a voting system between butterfly vectors pointing to the object’s center. The Butterfly detector is an anchor-free detector that overcomes the disadvantages of both anchor-based and anchor-free detectors by introducing the characteristics such as locating the center, width, and height information of the object of interest and generating the butterfly field from object-specific features with specific aspect ratios. The results demonstrated that this framework outperformed Faster R-CNN [83], CenterNet [84], and ClusDet [85] frameworks while improved performance than YOLOv3 [25], SSD [81], and Guided Attention Network (GANet) [86] frameworks for UAVDT dataset [45]. Furthermore, Zhang et al. [11] also used the VisDrone-DET dataset [87], to improve the performance in terms of different view points and angles by using ResNet-50 [88] as a backbone and Deformable Convolution Network (DCN) [89]. Moreover, as per [74], lack of depth features leads targets to be missed, while impediments such as shadows and trees induce false object detection. Therefore, in [74], with the Faster R-CNN framework [83], low-level features were coupled with high-level features utilizing Feature Pyramid Networks (FPN) to solve these difficulties. Also, anchor sizes and ratios were optimized.

Most datasets supplied for training contain only day time images. However, traffic surveillance often requires nighttime monitoring. Night time vehicle detection is more challenging due to artificial lights, road reflections, and darkness [75]. In this context, [75] proposed Faster R-CNN framework [83] with domain adaptation concept and using Generative Adversarial Networks (GANs), specifically cycle GAN approach [90], annotated daytime images successfully turned into unlabelled nighttime images. This study created two datasets: one with images, Day-time and Night-time Vehicle Detection (DNVD), and one with videos, Traffic Flow...
TABLE 4. Summary and comparison of DL-based vehicle detection systems using drone imagery data.

| Ref. | Framework | Backbone model | Dataset(s) | Image Resolution | Altitude | UAV/Camera state | Object class | Outcome |
|------|-----------|----------------|------------|------------------|----------|------------------|--------------|---------|
| [2]  | SA-CF     | ResNet-101 based attention model | i. CARPK; ii. PUCPR-PRA; iii. VisDrone2018-car; iv. UAVDT | 300 x 300 | Different altitudes | Different views | Single class | 1. 89.3  93.3  69.9  54.1 |
| [11] | Cascaded Framework | ResNet-50+DCNN | VisDroneDet2019 | 2000 x 1500 | Different altitudes | Different angles, Static and moving state of UAV | Ten classes | 22.61 |
| [68] | BoVW + CNN model | Inception-ResNet-v1 | VUC | -- | Different altitudes | Different views | Four classes (Sedan, SUV, Bus, Truck) | 95.25 |
| [69] | SPP-SSD | ResNet-101 | Custom | UAVDT | 3840 x 2160 | Different altitudes | -- | Three classes (Car, Bus, Truck) | 84.44  27.84 |
| [70] | YOLOv3 | DarkNet | DOTA | 800 x 800 or 44 x 44 | -- | -- | Fifteen classes | 69 |
| [71] | YOLOv3 with ISFOM | CNN | Custom | 3840 x 2160 or 1372 x 941 | Different altitudes | -- | One class (Car) | 88.78 |
| [72] | i. Faster R-CNN ii. YOLOv3 | Inception-ResNet-v2 | -- | 55m and 80m | -- | One class (Car) | -- | 88.38  99.94 |
| [73] | Butterfly | HRNet | VisDrone2019 | 2000 x 1500 | 1080 x 540 | Different altitudes | VisDrone 10 Classes | i. 90.15 ii. 58.1 |
| [74] | Modified Faster R-CNN | ResNet-101 | Custom | 9000 x 6700 | Fixed | Fixed | One class (Car) | 98.3 |
| [75] | Faster R-CNN with Domain Adaptation | VGG-16 | i. DNYD ii. TFPE | 1280 x 720 | -- | Fixed camera | One class (Car) | 54.6  87.8  57.6 (Day) |
| [76] | RT-SegRBM-Net | Encoder-Decoder Network | NPC_C, UAV_IR_DATA | 512 x 240 & 640 x 512 | 80 & 120m | Moving platform | One class (Vehicle) | 97 |

Parameter Estimation (TFPE). Another widely employed approach for drone-based traffic monitoring system in night scenarios consists in using thermal infrared imagery. In this context, to detect vehicles with semantic segmentation, a framework combining convolutional encoder/decoder and Gaussian-Bernoulli Restricted Boltzmann Machine (GB-RBM) networks was proposed by Salvo et al. [76]. The proposed model outperformed Mask R-CNN, a instance segmentation-based detection method.

2) VIDEO-BASED DETECTION

The shortlisted studies related to the traffic monitoring system based on drone videos are summarized in this section. Table 5 reports the video-based vehicle detection systems highlighting the different system characteristics (e.g., number of object classes, employed framework). Implementing vehicle detection tasks from videos with UAV motion is a critical task to deal with. Authors, in [12], [37], and [91] have developed the systems in the presence of UAV motions. Both static and moving states of UAV were considered while recording the videos with optical camera in [91], using Faster R-CNN framework [83] to detect the small and large vehicles. Since a moving UAV platform was employed, causing the detection of new vehicles in consecutive frames, additional techniques were also incorporated such as Feature-Based Image Alignment (FBIA) technique and Structural Similarity Index Measurement (SSIM). The former was employed on each frame to ensure the true object location while the latter was used to measure the similarity between consecutive frames. In this connection, the decision to perform detection is taken based on a threshold on the SSIM value. In addition, towards achieving the speed estimation of multiple vehicles in UAV videos, Li et al. [12] carried out vehicle detection using YOLOv3 [25] with a dataset obtained as a combination of custom and public UA-DETRAC [99] data. In this framework, vehicle occlusion issue was not experienced due to the high altitude of the UAV. On the other hand, small targets, complex background and motion blur effects were considered and dealt with during model training. Furthermore, Homography based motion estimation technique was employed to eliminate the background motion due to UAV camera movements. It is a transformation matrix and compensates for camera motion by mapping the points from one frame of a video sequence to the corresponding points on another frame [100]. Another interesting work is presented in [37], which implemented a Haar cascade classifier and CNN model to detect the vehicles in dense and dense-free traffic conditions from UAV videos including ego motion.

Towards traffic monitoring from UAV videos on highways, Brkić [92] used Faster R-CNN framework [83] with ResNet-50 backbone model [88] to detect the vehicles. The UAV was static and vertically positioned at the height of approximately 50 m. The UAV video was converted to frames, and image alignment and cropping of the monitored highway area were used for suitable image processing. Similarly, Zhang et al. [66] also used Faster R-CNN framework [83] for vehicle detection task in UAV videos. In addition, Zhang et al. [67] carried out traffic monitoring task in an urban area, and used Mask R-CNN framework [101] to detect the vehicles from videos recorded from a fixed positioned UAV. Different complex road scenes such as long vehicles shadows, vehicles with lights-on, roundabouts, interchange roads and different viewpoints were considered in [93] where YOLOv3 framework [25] was employed for the vehicle detection task. The performance of system was invariant to the orientation and scale variations in UAV videos. Similarly, [102] and [103] also used YOLOv3 framework [25] for traffic monitoring tasks, where [102] defined the Region of Interests (RoIs) at the starting and ending points of the monitoring area. Also, [103] applied detection to each frame of the UAV video, while emphasis was given on the vehicle counting task.
It is observed that in most of the studies, the length of the recorded videos is rather short, as long duration of videos could be subject to data loss and may require some sort of optimization approaches to maintain integer data. In addition to this, recording videos from UAVs do not necessarily imply top view of the vehicles for their surveillance, but also to acquire images of vehicles from their front, back and side views, thus making the traffic monitoring task quite challenging. Given such scenarios, Micheal and Vani [24] performed optimization by identifying key-frames using histogram-based approach and selected principal key-frames, thus reducing the video viewing time. The problem of vehicle detection from different views was covered by training the Faster R-CNN framework with data coming from multiple views. Two datasets were used: UAV123 [46] and VIVID [104]. In addition, problems of illumination changes and in-plane rotation were tackled in [53] while performing traffic monitoring tasks, showing that Faster R-CNN provided optimum result in comparison to background subtraction algorithm [18], frame difference approach [35], Viola-Jones method [105] and HoG+SVM approach [22].

The use of a pre-trained model, trained on images captured from a ground viewpoint such as MS-COCO [77] and ImageNet [56] datasets, leads to weak performance in the detection of vehicles from UAVs, due to the different characteristics of the captured images. In order to address this issue, [94] trained Deeply Supervised Object Detector (DSOD) learning model from scratch, entirely on UAV dataset rather than using a pre-trained model, showing performance improvement, both in terms of precision and recall. View point, scale and illumination variation effects were also considered in the training phase. Concerning the real-time detection and tracking of vehicles in aerial videos, [95] showed that YOLOv2 framework [82] provides higher accuracy and faster processing compared to YOLO [106] and SSD [81] frameworks. Further investigation on the training procedure and selection of specific parameters affecting the detection performance was provided by [96] concerning the YOLO framework [106]. Furthermore, to address the real-time monitoring and occlusion problems in traffic monitoring system, [107] used FPN [108] and observed high precision at the cost of high computational complexity when considering high resolution images. Focus was given on vehicle tracking to ensure real-time monitoring and to overcome the occlusion problem.

To investigate the impact of target size, number of training samples and combination of different datasets on the vehicle detection performance, [97] prepared the dataset by recording videos from UAVs in the presence of different environmental conditions such as lighting, congestion, surrounding and shadow. Experimental tests showed that the SSD framework [81] has good performance compared to Faster R-CNN [83] and Region-based Fully Convolutional Network (R-FCN) [109] detectors. Also, combining different datasets does not guarantee an improvement in performance. Furthermore, to obtain higher detection performance for small vehicles in UAV videos, [98] used RoI pooling layer on each convolution layer of VGG-16 backbone model and then concatenated the extracted multi-layer features.

Moreover, Li et al. [98] claimed that detecting vehicles in each video frame could be a waste of computational resources due to minor differences in successive frames. Also, detection of the same vehicles in successive frames could be difficult. Therefore, Li et al. [98] performed detection along with tracking approach, where detection is performed on n successive video frames and then used for region proposal. Additionally, Sun et al. [110] investigated the performance of Faster R-CNN [83] and SSD [81] detectors with MobileNet [111], GoogleNet/Inception V2 [112] and ResNet-50 [88] backbone models for UAV videos. It was observed that the detection rate was faster in SSD, but accuracy was compromised compared to Faster R-CNN framework. Also, concerning feature extraction accuracy, GoogleNet/Inception V2 provides more promising results than ResNet-50, while MobileNet shows poor performance when compared to both ResNet-50 and GoogleNet/Inception V2.
TABLE 6. Summary of UAV-based traffic monitoring systems including on-board processing.

| Ref. | Framework | Backbone model | Dataset(s) | Video/Image resolution | Altitude | UAV/Camera state | Hardware | Object class | Outcome |
|------|-----------|----------------|------------|------------------------|----------|------------------|----------|--------------|---------|
| [5]  | MoBIEYE  (YOLOv4) | CSPDarkNet53 Lite | Aeroscapes custom datasets | 3072 × 1728 | 20m to 150m | Isometric views (Moving Camera) | Nvidia Xavier NX | Multi-class | 83.40 | 90 |
| [9]  | CNN-based framework | ResNet-34 | Customs | 256 × 256 | — | — | UAV | Congestion/ Non-congestion | 93.50 | 95 |
| [113] | DnNeNet | Darknet | Customs | 4096 × 2160 | Any altitudes | Different view points | Odroid-XU4, Raspberry Pi 3 | One class (Vehicle) | 95 | |

B. ON-BOARD TRAFFIC MONITORING

Typically, data collected by drones and UAVs is transferred to servers and clouds for offline monitoring, surveillance, and analysis. On the contrary, in specific scenarios traffic monitoring is required for emergency responses, such as vehicle detection in search and rescue situations, vehicle accident identification, and traffic monitoring on busy roads. In this context, on-board (UAV) DL-based frameworks have been used to perform not only data collection but also real-time traffic surveillance. Unfortunately, a major drawback of approaches based on local processing lies in the fact that such solutions become computationally costly in terms of power consumption, flight duration and battery life.

The summary of explored on-board traffic monitoring systems is presented in Table 6. To match the computational cost of processing a video with DL frameworks on low-power and low-cost computing platforms such as UAVs/drones, [113] proposed a real-time on-board vehicle detection framework called DroNet, based on Tiny-YOLO [106]. The technique was made both efficient and lightweight, as well as computationally fast, by altering the architecture in terms of convolutional, maxpool, and detection layers, and number and size of filters. Both Odroid-XU4 and Raspberry Pi-3 modules achieved over 95% accuracy, although Odroid-XU4 proved faster in processing frames. Furthermore, Jian et al. [9] developed a real-time UAV-based traffic congestion detection system, where the vehicle recognition output was forwarded to a traffic monitoring center for further analysis. In this work authors decided not to recognize each vehicle separately, in order to allow for real-time traffic analysis and to address personal privacy concerns. Moreover, in the frameworks of real-time and on-board UAV traffic monitoring, Balamuralidhar et al. [5] designed a system using Enet [114] and YOLOv4 [26], in which CSPDarknet53 backbone model [115] was modified by reducing the amount and re-arranging convolutional layers and CSP Bottleneck blocks. This resulted in a lighter version of CSPDarknet, which at the same time caused a decrease in detection performance due to the increased amount of latent space related to the Bottleneck blocks. To mitigate this issue, the Space-to-Depth layer [116] was used to transform the input image into a reduced spatial dimension, while enhancing depth.

C. PRE-PROCESSING APPROACHES

Using image pre-processing and augmentation techniques help deep neural network models to learn better. Since the training process works by extracting features from provided input images, these approaches help the model to learn by presenting the same scene from diverse perspectives. Different state-of-the-art techniques used to train the DL model for the vehicle detection task are discussed in the following. Concerning image pre-processing, in [70] all input images were tiled into 512 × 512 pixels and tiles containing no object were not considered in the training part. Similarly, in [71] images are also divided into patches, where only areas containing objects were selected using SURF algorithm [19] for the training purpose. Furthermore, Wang et al. [74] divided high resolution images of 9000 × 6700 pixels, taken by drone with fixed height and angle, into 100 pieces and cropped to 900 × 670 pixels, also discarding the duplicates before training. Moreover, to improve multi-scale vehicle detection accuracy, Li et al. [69] reduced the image sizes without distorting the vehicle information by cropping and dividing the raw input image into two patches. These two patches and the original image were combined together into a single batch to feed CNN for feature extraction. In addition, in [11] the input images were segmented into a 4 × 4 block, adding the segments together with the original image, thus increasing the training set by 5 × times. This practice of data augmentation helped in terms of detection of small size objects using the aerial images.

Due to instability of recorded video, image alignment techniques (feature descriptors, homography between images) were employed in [92]. For feature descriptors and homography, Oriented FAST and Rotated BRIEF (ORB) and RANdom SAmple Consensus (RANSAC) algorithms were used for their implementation. Due to memory and processing speed constraints, original frame dimensions were reduced from 4096 × 2160 to 3797 × 400 pixels by cropping, before feeding to the detection framework. Furthermore, concerning the data preparation task in [95], circulant structure of tracking-by-detection with kernels [117] was used to annotate the training images of a video scene, while rotation based data augmentation technique was used to increase the number of training samples. Another interesting work in this framework is [96], where augmentation techniques (random scaling and translation) and dropout (0.5 rate) were implemented in the training phase in order to avoid over-fitting. Furthermore, horizontal and clockwise rotations were performed as augmentation techniques before training in [98].

Given the above analysis concerning image pre-processing techniques for high resolution aerial images, it is observed that augmentation techniques such as dividing the original
image into patches are often used. Although integrating original images with their patches increases computational load, the final aim is to improve the overall vehicle detection performance by artificially increasing the available training dataset.

**D. TRAFFIC OBJECT COUNTING**

When an object detector is used to detect the vehicles in each frame of a video, and if the counting by detection approach is employed, then there are chances of redundancy if the same vehicle appears in consecutive frames. For instance, if we consider a video sequence lasting 3 s and divide it into 3 frames, where each frame contains the same 3 vehicles, applying a counting mechanism on each frame separately would result in counting a total of 9 vehicles instead of 3. This redundancy is unwanted and must be tackled when designing traffic vehicle counting systems.

A summary of tasks and techniques used in the literature for vehicle counting are presented in Table 7. For what concerns UAV videos, counting vehicles and minimizing the redundancy in successive frames is achieved in [103] by identifying the same vehicle in successive frames based upon specific features, and then counting the single vehicles. The orientation/shape, texture, and color features of the detected vehicles are computed using the HoG, LBP, and Mean RGB feature extractors, respectively. All the obtained features are used to build a single feature vector describing the vehicles inside consecutive frames. An Euclidean distance-based similarity measure between the two resulting feature vectors is determined by setting a threshold value, and if the obtained similarity score is greater than the specified threshold value, then a new vehicle is counted.

Detection lines and virtual loops, as mentioned in [118], have been used in the literature to count vehicles in a video sequence in addition to the counting framework discussed above. The former approach is suitable for counting the fast-moving vehicles, whereas the latter is better for slow-moving vehicles, especially on congested routes. In the detection line solution a virtual line is drawn across the road, which splits the two sides of a lane. In this way, a vehicle passing through the road would always intersect the virtual line, hence increasing the vehicle count. The detection line approach is employed in works such as [92] and [67]. Concerning loop-based approach, a virtual loop is drawn at the bottom center of each lane of a road, with the same length of the lane itself. Every virtual loop has a vehicle status flag, to indicate the vehicles, that is determined by computing the ratio of the vehicle pixels with respect to all pixels and the average width of the objects inside the loop.

Tracking based counting of vehicles is carried out in [3] and [66], where Zhu et al. [3] proposed Deep Vehicle Counting Framework (DVCF) to count the vehicles from high resolution (3840 × 2178 px) UAV videos. The number of tracker was assigned to each detected vehicle in the initial frame, and it was increased or decreased for newly entered or disappeared vehicles based on matching the vehicles in successive frames. Further, Ke et al. [37] and Li et al. [75] count the detected vehicles in videos by estimating their motion. In [37], after vehicle detection, vehicle count was updated on the basis of trackers obtained by motion estimation, by employing Kanade-Lucas-Tomasi (KLT) method [119] with optical flow. A different approach was used in [2], where authors conducted simultaneous detection and counting in UAV imagery by introducing a counting layer after the feature extractor in the CNN framework.

In addition to these studies, counting of objects in crowd is implemented using DL based Congested Scene Recognition (CSR) concept, also referred as density map estimation, in [120] which is not only applied on people but have been proven for vehicles too. The idea in CSRNet [120] is to use the deeper CNN network, end-to-end trainable, to extract the high level features and produce the heat maps for counting. Further, according to [121] density maps represent ambiguous features of objects in congested situations, resulting in an increase in error. As a result, object counting is decoupled into probability and count map regressions, with the former illustrating the probability of each pixel being an object and the latter counting the objects based on the probability map regression. Furthermore, the frequency feature pyramid module is implemented in [122] to address the issues of scale variations, imbalanced data distribution, and insufficient local features in counting crowds and vehicles. This module addresses the multi-scale variation and imbalance data distributions using frequency branches and the global-local consistency loss function, respectively.

**E. TRAFFIC OBJECT TRACKING**

Objects tracking is especially useful when object detection is challenging due to various factors such as motion blur, occlusion, scale and angle variations. In a broader view, tracking algorithms can be classified as online/offline, SOT/MOT, and detection-based/detection-free. In SOT approaches, objects are tracked through their entire motion, independently of their detection, whereas in MOT methods, objects are tracked only if they are first detected and localized. For this reason, MOT is considered a detection-based tracking approach. Furthermore, MOT solutions employ both offline and online approaches, where offline methods achieve better performance, while online solutions prove more robust.

Various approaches to object tracking have been proposed in the literature, such as Intersection over Union (IoU)-based tracking [123], Simple Online and Real-Time Tracking

**TABLE 7. Summary and comparison of vehicle counting solutions based on drone images/videos.**

| Ref. | Task | Approach | Outcome (Accuracy, %) |
|------|------|----------|-----------------------|
| [3]  | To count vehicles in high resolution videos | Counting-by-tracking | 93.7 |
| [7]  | To get traffic density | Motion estimation based counting | 90.4 |
| [75] | To measure the traffic density | Lane-based counting | 55.2 (Night time) 97.38 (Day time) |
| [103] | To obliterate the same vehicles | Similarity matching using Euclidean Distance | 52.8 |
ent filtering-based techniques have been used to track the approach. In this framework, differ-
tracking-by-detection
ences are made to determine the lines between two adjacent frames. If fast moving vehicles pass during light traffic conditions, the threshold value will be high, whereas in the event of slow
tracking-for-the-same-target-were-generated-by-joining
over, the Euclidean distance information was used in [66]
parameters-collected-from-the-instance-segmentation. More-
to-handle-the-occlusion-problem,-but-it-also-can-not-provide-robust
DCF-tracker-alone-is-not-only-unable-to-handle
performing-vehicle-Re-IDentification
KF technique. However, in [12] the main focus was given on multiple vehicles speed estimation by taking into account tracking and motion estimation. From a different perspective, in [93] vehicle tracking is achieved by performing vehicle Re-IDentification (Re-ID) with deep features and motion estimation with KF technique. As stated in [107], DCF tracker alone is not only unable to handle the occlusion problem, but it also cannot provide robust tracking. Therefore, a combined model of KF and DCF is used in [107], which is more robust and reliable than individual DCF tracker. However, according to [37], KF and particle filtering techniques are not suited for vehicle tracking from UAV videos with background motion. To address this problem, Ke et al. [37] employed the KLT approach [119], an interest point method, to estimate background and vehicle motions. Concerning object data association, which is the problem of matching the predicted BBoxes of existing vehicles with the detected BBoxes of vehicles in the current frame, so as to optimize the number of matches in the two sets of BBoxes, a Hungarian algorithm [129] is implemented in [3], [12], and [107].

The MOSSE tracker is employed in [5], which is an accurate and fast algorithm, where estimated correlation filters are used to approximate the detected objects location in a video frame to track the detected vehicles [127]. During tracking, the area of the detected object in the first frame is transformed into frequency domain using Discrete Fourier Transform (DFT) to generate the synthetic data for the initialization of tracker and update of the filter. Furthermore, in order to track the vehicles with a moving UAV, [91] used DCF filter with Channel and Spatial Reliability Tracking (CSRT), where CSRT was employed due to its ability to achieve high accuracy with respect to MOSSE. Two standard features, HoG and color names, have been used in CSRT and the decision to start tracking is dependent upon a threshold on SSIM. Moreover, Kernel Correlation Filter (KCF) algorithm [130] was employed for tracking in [98].

In addition to these algorithms, [92] and [102] implemented

Table 8 provides an overview of the main vehicle tracking frameworks used in traffic monitoring.

1) TRACKING-BY-DETECTION
This approach involves target detection on each successive frame and then applying data association to match the objects. However, if a DL model takes a long time to detect the objects, this approach will likely not satisfy real-time constraints. Therefore, Fernandez-Sanjurjo et al. [107] used a different strategy in which detection was performed a predefined number of times, taking into account a real-time scenario. Furthermore, in [67], instance based tracking was used, where vehicles were detected in subsequent frames based on parameters collected from the instance segmentation. Moreover, the Euclidean distance information was used in [66] in order to distinguish new vehicles based on a particular threshold value, depending on the motion speed. For example, if fast moving vehicles pass during light traffic conditions, the threshold value will be high, whereas in the event of slow moving vehicles, the threshold value will be smaller. Finally, motion tracks for the same target were generated by joining the lines between two adjacent frames.

2) ONLINE AND FILTERING BASED TRACKING
Most of the techniques discussed here are based on the tracking-by-detection approach. In this framework, different filtering-based techniques have been used to track the vehicles. KF technique is implemented in [24] to track the vehicles in a UAV sequence, where vehicle’s current state is predicted based upon the motion model of the last state and then vehicle’s position is updated by leveraging its centroid information. On the same line, [3] and [12] have also employed KF technique. However, in [12] the main focus was given on multiple vehicles speed estimation by taking into account tracking and motion estimation. From a different perspective, in [93] vehicle tracking is achieved by performing vehicle Re-IDentification (Re-ID) with deep features and motion estimation with KF technique. As stated in [107], DCF tracker alone is not only unable to handle the occlusion problem, but it also cannot provide robust tracking. Therefore, a combined model of KF and DCF is used in [107], which is more robust and reliable than individual DCF tracker. However, according to [37], KF and particle filtering techniques are not suited for vehicle tracking from UAV videos with background motion. To address this problem, Ke et al. [37] employed the KLT approach [119], an interest point method, to estimate background and vehicle motions. Concerning object data association, which is the problem of matching the predicted BBoxes of existing vehicles with the detected BBoxes of vehicles in the current frame, so as to optimize the number of matches in the two sets of BBoxes, a Hungarian algorithm [129] is implemented in [3], [12], and [107].

The MOSSE tracker is employed in [5], which is an accurate and fast algorithm, where estimated correlation filters are used to approximate the detected objects location in a video frame to track the detected vehicles [127]. During tracking, the area of the detected object in the first frame is transformed into frequency domain using Discrete Fourier Transform (DFT) to generate the synthetic data for the initialization of tracker and update of the filter. Furthermore, in order to track the vehicles with a moving UAV, [91] used DCF filter with Channel and Spatial Reliability Tracking (CSRT), where CSRT was employed due to its ability to achieve high accuracy with respect to MOSSE. Two standard features, HoG and color names, have been used in CSRT and the decision to start tracking is dependent upon a threshold on SSIM. Moreover, Kernel Correlation Filter (KCF) algorithm [130] was employed for tracking in [98].

In addition to these algorithms, [92] and [102] implemented

Table 8. Summary and comparison of vehicle tracking solutions using drone images/videos.

| Ref. | Task | Approach | Algorithm/technique | AP/mAP (%) | MOTA (%) | MOTP (%) | Precision (%) |
|------|------|----------|---------------------|------------|----------|---------|---------------|
| 5    | To achieve real-time MOT tracking with high frame rates | Filtering based | MOSSE Tracker | --         | 90.91    | 92.11    | --            |
| 24   | To track the object automatically in optimized UAV video. | Tracking-by-detection | KF | 96.03 | -- | -- | -- |
| 93   | To minimize the identity switch in MOT | Tracking-by-detection | KF, Vehicle Re-ID | -- | 81.30 | -- | -- |
| 94   | To cover the irregularities due to undetected vehicles. | Training based | LSTM | -- | -- | -- | 96.13 |
| 98   | To track vehicle in successive frames | Tracking-by-detection | KCF | 67.40 | -- | -- | -- |
| 107  | To deal the occlusion in real-time tracking | Filtering based | KF + DCF tracker | -- | 86.95 | 85.83 | -- |
the SORT tracking algorithm to track the vehicles in UAV videos.

3) TRAINING-BASED TRACKING
The LSTM framework [128] is utilized in [94] to track the vehicles, and it was trained based on the vehicle’s visual features and BBox information. To learn the sequential model and spatially predict the location of vehicles, LSTM leverages spatial and temporal information. When compared to KF-based tracking, LSTM shows improved tracking performance.

VI. DISCUSSION AND RESEARCH TRENDS: RQ3
RQ3 is addressed by discussing the information reported in Table 4, 5, 7, 8 and 6, concerning detection in images, videos, counting, tracking, and on-board systems, respectively. The performance of these systems, as well as the main challenges or limitations are analyzed and discussed in the following.

A. DISCUSSION
1) TRAFFIC OBJECT DETECTION
As shown in Table 4, 5, and 6, traffic monitoring systems have been evaluated using both state-of-the-art and custom-built datasets. These data not only compare implemented frameworks and their performance, but also provide information about the features of the UAV platform, such as altitude, UAV/camera status, and resolution used to capture aerial data. Although it is more reasonable to acquire UAV data from bird’s eye view in order to obtain accurate traffic information, in some places privacy concerns force UAVs to be piloted at high altitudes or from side views, which may result in performance reduction due to differences in shooting angles. The majority of researchers consider various viewing angles, as noted in Table 4 and 5. Additionally, multiple altitude values ranging from 5 – 126 m have been considered when gathering data from various perspectives. It is important to notice that the exact value of altitude is not always reported in literature studies. It is also observed that both one- and two-stage detectors have been used, and the most widely employed frameworks are Faster R-CNN and YOLOv3 detectors, belonging the two- and one-stage categories respectively.

Moreover, it is also observed from Table 4 and 5 that object detectors achieved lower performance on publicly available datasets such as Visdrone and DOTA. However, the majority of the systems designed with custom prepared dataset achieved performance greater than 80%. This might also be due to the fact that most of them are application and area specific, considering a limited number of object classes, thus resulting in a less complex detection task. Indeed, as it can be noted from Table 4 and 5, these systems considered up to four target classes. For instance, Li et al. [75] prepared a custom dataset with only one class, obtaining a detection performance of 87.99%. It is reasonable to assume that an increase in the complexity level caused by considering other issues, such as a larger number of classes, higher altitudes and different lighting conditions, would likely degrade the overall system performance. In addition, in the majority of vehicle detection systems DL models are trained using weights of pre-trained models based on large scale datasets such as MS-COCO dataset [77], which contains regular and high resolution images in their natural context. On the contrary, drone images have diverse and irregular viewpoints, small and dense scenarios. Therefore, the use of weights trained using drone-based large scale imagery would produce a performance boost [94]. A further challenge is related to target size. As altitude increases, vehicles size gets smaller. Also, with continuous variations in altitude, the corresponding impact is observed on object sizes.

In addition, simultaneous variations in altitude and movement of vehicles make multi-scale detection a quite challenging task. Although researchers have tried to tackle this crucial and critical problem, in most cases they have not explicitly mentioned the considered altitude information. In this connection, Ham et al. [97] suggested that the SSD framework is promising for vehicle detection task in aerial imagery and its inherent multi-scale feature structure facilitated multi-size object detection, making this solution suitable for traffic monitoring at generically high altitudes. However, the exact value or range of UAV altitude considered by the authors was not mentioned for the obtained results. In addition, results produced by Zhu et al. [3] inferred that Enhanced-SSD works better than SSD, Faster R-CNN and YOLO frameworks for the traffic monitoring tasks for specific urban road traffic scenarios recorded in UAV videos. Furthermore, according to [98], multi-level feature fusion is effective to increase the detection performance, especially when small objects are considered.

In most cases, it has been observed that the output of vehicle detection is obtained in terms of BBox for each detected vehicle. However, more precise information about detection can be obtained by further segmenting each identified vehicle. The framework proposed in [76] achieved better results in segmenting the targeted vehicles for thermal images when compared to existing segmentation-based frameworks such as Mask R-CNN detector [101]. However, this framework considers only a single class, while performance may vary when dealing with a multi-class scenario. Furthermore, based on various experimental analyses for object detection in UAV videos in [110], it was deduced that the selection of feature extraction model and detection framework must take into account the processing speed and accuracy, as well as the ability to detect objects of varying sizes from aerial views. Additionally, system configuration and memory size play a significant role in processing speed. As demonstrated in [91], while estimating the speed of vehicles in video captured with both static and moving UAV, the frame rate decreases as the number of vehicles increases, and vice versa.

According to the literature related to on-board system deployment, the YOLO framework and Darknet backbone
model are the most frequently used in case of on-board edge computing solutions to the purpose of traffic monitoring (see Table 6). This is driven by the need to create a lightweight model, while at the same time preserving accuracy and processing speed to meet the requirements of real-time scenarios. For instance, a lightweight and real-time efficient algorithm for edge computing has been devised by modifying the backbone structure, as stated in [5], in which the original CSPDarknet-53 was transformed into CSPDarknet-53 (Lite) to improve system performance in terms of speed and accuracy.

2) TRAFFIC OBJECT TRACKING AND COUNTING
Vehicle counting tasks have been performed in traffic monitoring systems to determine the number of vehicles, traffic density estimation, traffic flow rate, and traffic congestion analysis, as shown in Table 7. The accuracy of vehicle count estimation is strongly dependent on detector and tracking accuracy, as the count is computed using the number of detected objects in each frame or the number of motion trackers in most of the designed systems. The counting performance is reported either in terms of counting accuracy or in terms of the number of counted vehicles. Table 7 shows a counting accuracy exceeding 90%, where a custom dataset is utilized, thus achieving high performance. A primary challenge in vehicle detection lies in the lighting conditions. These are subject to vary in relation to the weather, moving or occluded light sources, different times of the day. These factors deeply influence the traffic monitoring capabilities of the systems. According to [75], the performance of detecting and counting vehicles at night is lower than during daytime, as it can be noted from Table 4. This is due to the fact that the complexity of recognizing vehicles increases at night due to diverse lighting conditions and the counting performance might be degraded due to occlusions.

Concerning the MOT task, various issues have been considered when designing the tracking section of traffic monitoring systems, including the association of BBoxes from different vehicles, vehicle matching in long duration videos, vehicle tracking with a moving UAV, identity switching minimization, and different levels of occlusion. The extracted data for the vehicle tracking task is shown in Table 8, along with the techniques and algorithms used to complete the task. It is noticed that the tracking outcome is depicted either separately in terms of MOTA and MOTP, or in conjunction with detection task in terms of AP. Additionally, the tracking outcome is illustrated by drawing motion trackers. Further, similarly to the counting performance, the performance of the tracking task is also high, greater than 80% in the majority of studies, as illustrated in the Table 8. The outcome of the available studies, however, is not directly comparable because each developed system took into account different and unique factors such as the type and characteristics of the considered dataset and the number of recognized classes.

B. FUTURE TRENDS
Some major issues and potential future trends in drone-based traffic monitoring systems from a DL perspective are discussed in the following.

1) TRAFFIC OBJECT DETECTION
State-of-the-art detectors such as YOLOv5 [131], YOLOX [132], and YOLOv6 [133] detectors, are worth investigating towards increasing the performance of traffic monitoring systems in case of both small and large scale objects. Also, the mentioned variants of the YOLO family are more suitable for processing a large number of frames per second with respect to other existing solutions. Therefore, investigating drone based traffic monitoring in this sense could be promising in terms of accuracy and real-time performance. Other interesting aspects requiring further research are pre-processing techniques such as Image alignment, which are not yet able to handle the issues of UAV elevation and angle variations precisely. In this connection, the joint use of georeferenced UAV videos may help in tackling alignment issues.

The majority of methodologies covered in this review study process UAV videos using static object detection frameworks rather than standard video object detection such as [134], [135], [136]. However, few of them have used image alignment techniques prior to detection, such as FBIA, in conjunction with static object detection frameworks, and the temporal information of the UAV videos has contributed in tracking task. Therefore, traffic monitoring systems can be implemented using typical video object detection techniques that make use of temporal and contextual information to address missed detection in consecutive frames. Thus, a comparative analysis of traffic monitoring system using static and video object detection frameworks may be conducted with respect to computational burden, accuracy and frame rate. A review of state-of-the-art video object detection solutions is presented in [137], and the details about spatio-temporal models and feature extraction strategies from DL perspective are given in [138].

Furthermore, implementing vehicle monitoring frameworks working in diverse weather and light conditions is also a future direction to pursue. Additionally, the detection of vehicles in aerial scenes during nighttime poses additional difficulties and challenges in terms of scarce lighting conditions, dark environment, high motion blur. Towards solving these problems, detection approaches with event based camera images could be more promising, compared to frame based detection approaches. Indeed, event-based solutions are invariant to absolute illumination levels and more robust to high latency and motion blur issues as reported in [139]. Event cameras, which provide high temporal precision, high dynamical coverage, and low data rates, represent a paradigm shift from conventional frame-based cameras. thanks to these properties, event cameras are especially well-suited for situations involving a lot of motion, and difficult lighting
conditions [139], [140]. Therefore, the design of specific frameworks for vehicle detection with event cameras during nighttime could be a relevant research topic to tackle open issues.

In the literature, the implementation of domain adaption for traffic monitoring systems is very rare. However, with domain adaptation, features from neural networks trained on a large open dataset can be used to perform the traffic monitoring task for some small dataset for a specific region or application. To this aim, similar types of data must be considered. In this connection, domain adaptation is used to train a neural network on one dataset (source with labels) and then test it on another dataset (target with no labels). Most of the time, it is difficult to gather diverse data using drones, or in some cases performing data annotation for the collected information becomes laborious. In such situations, the concept of domain adaptation can be helpful for the specific purpose of traffic monitoring.

In addition, type and shape of vehicles are different with respect to world region and location. For example, data in VisDrone dataset is captured in the streets of China, on the contrary, the MTID dataset [42] covers the traffic in European region. Besides including different types of commercial vehicles, they also present differences in labeling, even for similar types of object classes. Furthermore, although some drone-based aerial datasets are already available in the literature, such as VisDrone2021 [87] and DOTA [45] datasets, a drone-based large scale diverse aerial dataset in the context of traffic monitoring systems, equivalent to the standard MS-COCO dataset, is still missing. In this view, the use of individual datasets combined together to make a large scale dataset makes correct classification a quite complex task, since it may occur that similar types of vehicles are assigned different labels.

From a different perspective, to get the traffic monitoring output in terms of segmentation, especially real-time instance segmentation, object detection models such as Yolact [141], Yolact++ [142], and Yolactedge [143] can also be used to test vehicle monitoring in multi-class with heterogeneous data. Moreover, 3D vehicle detection in ground-captured RGB images is being explored recently for the purpose of autonomous driving by applying 2D scale first due to maturity of 2D object detection. Although, the traffic monitoring task is mainly performed on the 2D scale, applicability of 3D scale object detection for vehicles detection in aerial imagery can be investigated. This task is particularly useful in case of side view images captured from drones. Concerning real-time and fast processing scenarios, backbone models such as MobileNet [111], are classified as lightweight algorithms, and can be deployed on-board for edge computation. Also, lightweight frameworks such as YOLOv5-nano [131] can also be deployed for the same purpose. An interesting study in this framework might concern the performance analysis of various on-board deployed lightweight algorithms in terms of accuracy and processing speed for traffic monitoring.

2) TRAFFIC OBJECT TRACKING AND COUNTING

From traffic monitoring perspective, the goal of MOT is to track all the vehicles in an aerial sequence. In most cases, vehicles are of the same type, especially cars and vans. Also, if drone is at high altitude or traffic is dense, this causes problems such as occlusions, moving camera and different level of occlusions depending upon the camera viewpoint. Tackling all these issues with a single algorithm is a quite challenging task. Moreover, tracking algorithms based upon the concept of sparse representation of targets could be useful in the presence of different influential parameters such as motion blur, occlusion, low resolution, illumination variation, scale variation and background clustering [4]. An ad-hoc investigation on the use of such algorithms for multi-vehicle tracking could be an interesting research topic in the framework of UAV-based traffic monitoring.

The literature includes a large number of vehicle tracking systems implemented using the tracking-by-detection approach, where data association and motion models have been used in consecutive frames for the tracking task, given a target detection output. Any error in the detection output would reflect on the performance of the tracking task. Concerning this issue, [144] proposes as alternative tracking-by-regression, which can also be used for multiple-vehicle tracking to improve the overall system performance. In this approach, regression head of the object detector is used to compute the tracking task. The main issues related to the tracking problem are occluded vehicles, dense traffic conditions, sudden drone camera motions, large displacements due to low frame rates, small size of vehicles due to high altitudes, recovering the same vehicles from occlusions in the dense and high altitude scenes. In this connection, vehicle Re-ID and motion models in conjunction with tracking by regression might help to mitigate the aforementioned problems. In details, problems such as Re-ID of partially occluded vehicles and tracking multiple vehicles in a drone-based video sequence with high frame rates could strongly degrade the tracking performance. Additionally, since vehicles move quickly and sometimes appear to be identical (e.g., color and shape), the switch ID problem is a significant concern. In a nutshell, tackling all these issues for the MOT task in traffic monitoring systems is an attractive and current research trend.

On one hand, a thorough analysis of the state-of-the-art showed that the different traffic monitoring tasks, namely detection, counting and tracking, have been usually carried out separately. On the other hand, the output of one task is commonly used as input to facilitate the subsequent. For instance, the detection output is used as input to the tracking task and the outcome of tracking helps to perform the counting task. However, a DL-based framework can be designed to carry out all these tasks jointly, by defining a new loss function for the network training. The architecture of such a framework would reflect in merging the tracking and counting blocks along with the detection block in order to devise a complete DL-based vehicle detection framework.
VII. CONCLUSION

The systematic review reported in this paper provides a thorough overview of the state-of-the-art contributions in the framework of drone-based traffic monitoring from a DL perspective. The application of UAVs in the field of traffic monitoring has been studied and classified, putting particular emphasis on vehicle identification, tracking, and counting. The design of traffic monitoring systems for different and diversified drone datasets heavily relies on DL algorithms and image processing. As a result, the extraction of specific relevant information proves critical in achieving good performance in vehicle tracking, traffic density/congestion analysis. As a general observation, the most common approach to vehicle detection consists in using static object detection frameworks; also, vehicle counting is usually strictly dependent upon the performances of prior detection and tracking. Furthermore, systems trained by employing custom datasets achieve higher accuracy with respect to those based on the use of public datasets. Finally, in this survey we provide a reasoned discussion, analyzing and highlighting possible future research trends driven by the main issues and open problems identified in the state-of-the-art that can lead to optimal traffic monitoring solutions. For instance, a drone-based, large-scale, diverse aerial dataset comparable to the standard MS-COCO dataset is still lacking, which could be advantageous in terms of performance gain when designing drone-based surveillance systems utilizing DL frameworks.

REFERENCES

[1] F. Outay, H. A. Mengash, and M. Adnan, “Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges,” Transp. Res. A, Policy Pract., vol. 141, pp. 116–129, Nov. 2020.
[2] W. Li, H. Li, Q. Wu, X. Chen, and K. N. Ngan, “Simultaneously detecting and counting dense vehicles from drone images,” IEEE Trans. Ind. Electron., vol. 66, no. 12, pp. 9651–9662, Dec. 2019.
[3] J. Zhu, K. Sun, S. Jia, Q. Li, X. Hou, W. Lin, B. Liu, and G. Qiu, “Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 12, pp. 4968–4981, Dec. 2018.
[4] M. Wan, G. Gu, W. Qian, K. Ren, X. Maldague, and Q. Chen, “Unmanned aerial vehicle video-based target tracking algorithm using sparse representation,” IEEE Internet Things J., vol. 6, no. 6, pp. 9689–9706, Dec. 2019.
[5] N. Balamuruladhar, S. Tilon, and F. Nex, “MultiEYE: Monitoring system for real-time vehicle detection, tracking and speed estimation from UAV imagery on edge-computing platforms,” Remote Sens., vol. 13, no. 4, p. 573, Feb. 2021.
[6] T. Moranduzzo and F. Melgani, “Detecting cars in UAV images with a catalog-based approach,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 10, pp. 6356–6367, Oct. 2014.
[7] G. Salvo, L. Caruso, and A. Scordo, “Urban traffic analysis through an UAV,” Proc. Social. Behav. Sci., vol. 111, pp. 1083–1091, Feb. 2014.
[8] K. Masood and M. A. Khan, “Trajectory tracking and hurdle avoidance for an autonomous quadrotor UAV,” Eur. J. Adv. Eng. Technol., vol. 4, no. 2, pp. 135–142, 2017.
[9] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon, “Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads,” IEEE Consum. Electron. Mag., vol. 8, no. 3, pp. 81–86, May 2019.
[10] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, and A. Sciarone, “Drone surveillance system—RF/WiFi-based drone detection, localization, and tracking: A survey,” in Avionics Cybersecurity: Foundations, Principles, and Applications. Edison, NJ, USA: IET, 2021, pp. 49–81.
[11] X. Zhang, E. Izquierdo, and K. Chandramouli, “Dense and small object detection in UAV vision based on cascade network,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 1–9.
[12] J. Li, S. Chen, F. Zhang, E. Li, T. Yang, and Z. Lu, “An adaptive framework for multi-vehicle ground speed estimation in airborne videos,” Remote Sens., vol. 11, no. 10, p. 1241, May 2019.
[13] United Nations. (2019). World Population Prospects 2019, Accessed: Sep. 19, 2021. [Online]. Available: https://www.un.org/development/desa/publications/world-population-prospects-2019-highlights.html
[14] M. A. Khan, W. Ectors, T. Bellemens, D. Janssens, and G. Wets, “UAV-based traffic analysis: A universal guiding framework based on literature survey,” Transp. Res. Proc., vol. 22, pp. 541–550, Jan. 2017.
[15] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, “Video processing techniques for traffic flow monitoring from aerial survey,” in Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC), Oct. 2011, pp. 1103–1108.
[16] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, “Hierarchical and networked vehicle surveillance in its: A survey,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 1, pp. 25–48, Jan. 2017.
[17] J. Zhou, D. Gao, and D. Zhang, “Moving vehicle detection for automatic traffic monitoring,” IEEE Trans. Veh. Technol., vol. 56, no. 1, pp. 51–59, Jan. 2007.
[18] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” IEEE Trans. Image Process., vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[19] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2006, pp. 404–417.
[20] K. Mu, F. Hui, and X. Zhao, “Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching,” J. Inf. Process. Syst., vol. 12, no. 2, pp. 183–195, 2016.
[21] S. Han, Y. Han, and H. Hahn, “Vehicle detection method using Haar-like feature on real time system,” World Acad. Sci., Eng. Technol., vol. 59, no. 35, pp. 455–459, 2009.
[22] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2005, vol. 1, pp. 886–893.
[23] Z. Dong, Y. Wu, M. Pei, and Y. Jia, “Vehicle type classification using a semisupervised convolutional neural network,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 4, pp. 2247–2256, Aug. 2015.
[24] A. A. Micheal and K. Vani, “Automatic object tracking in optimized UAV video,” J. Supercomput., vol. 75, no. 8, pp. 4986–4999, Aug. 2019.
[25] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv:1804.02767, 2018.
[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.
[27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[28] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, “Mask R-CNN,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, Jun. 2017.
[29] M. Zohaib, M. Taimana, M. G. Padalkar, and A. D. Bue, “3D key-points estimation from single-view RGB images,” in Proc. Int. Conf. Image Anal. Process. Cham, Switzerland: Springer, 2022, pp. 27–38.
[30] I. Bisio, H. Haleem, C. Garibotto, F. Lavagetto, and A. Sciarone, “Performance evaluation and analysis of drone-based vehicle detection techniques from deep learning perspective,” IEEE Internet Things J., vol. 9, no. 13, pp. 10920–10935, May 2022.
[31] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 408–417.
[32] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, “Methods of analyzing traffic imagery collected from aerial platforms,” IEEE Trans. Intell. Transp. Syst., vol. 4, no. 2, pp. 99–107, Jun. 2003.
[33] X. Cao, J. Lan, P. Yan, and X. Li, “Vehicle detection and tracking in airborne videos by multi-motion layer analysis,” Mach. Vis. Appl., vol. 23, no. 5, pp. 921–935, Sep. 2012.
[34] H. Yalcin, M. Hebert, R. Collins, and M. J. Black, “A flow-based approach to vehicle detection and background mosaicking in airborne video,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, Jun. 2005, p. 1202.
[35] A. C. Shastry and R. A. Schowengerdt, “Airborne video registration and traffic-flow parameter estimation,” IEEE Trans. Intell. Transp. Syst., vol. 6, no. 4, pp. 391–405, Dec. 2005.
[83] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[84] L. W. Sommer, T. Schuchert, and J. Beyerer, “Fast deep vehicle detection in aerial images,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2017, pp. 311–319.

[85] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, “Clustered object detection in aerial images,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 8311–8320.

[86] Y. Cai, D. Du, L. Zhang, L. Wen, W. Wang, Y. Wu, and S. Lyu, “Guided attention network for object detection and counting on drones,” 2019, arXiv:1909.11376.

[87] Y. Cao et al., “VisDrone-DET2021: The vision meets drone object detection challenge results,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2021, pp. 1–14.

[88] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[89] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 764–773.

[90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2223–2232.

[91] D. Bisswas, H. Su, C. Wang, and A. Stevanovic, “Speed estimation of multiple moving objects from a moving UAV platform,” ISPRS Int. J. Geo-Inf., vol. 8, no. 6, p. 259, May 2019.

[92] I. Brikic, M. Miler, M. Ševrovic, and D. Medak, “An analytical framework for accurate traffic parameter calculation from UAV aerial videos,” Remote Sens., vol. 12, no. 22, p. 3844, 2020.

[93] J. Wang, S. Simeonova, and M. Shahbazi, “Orientation- and scale-invariant multi-vehicle detection and tracking from unmanned aerial videos,” Remote Sens., vol. 11, no. 18, p. 2155, Sep. 2019.

[94] A. A. Micheal, K. Vani, S. Sanjeevi, and C.-H. Lin, “Object detection and tracking with UAV data using deep learning,” J. Indian Soc. Remote Sens., vol. 49, pp. 463–469, Nov. 2020.

[95] T. Tang, Z. Deng, S. Zhou, L. Lei, and H. Zou, “Fast vehicle detection in UAV images,” in Proc. Int. Workshop Remote Sens. Intell. Process. (RSIP), May 2017, pp. 1–5.

[96] M. Radovic, O. Adakdwa, and Q. Wang, “Object recognition in aerial images using convolutional neural networks,” J. Imag., vol. 3, no. 2, p. 21, 2017.

[97] S. W. Ham, H.-C. Park, E.-J. Kim, S.-Y. Kho, and D.-K. Kim, “Investigating the influential factors for practical application of multi-class vehicle detection for images from unmanned aerial vehicle using deep learning models,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2674, no. 12, pp. 553–567, Dec. 2020.

[98] S. Li, W. Zhang, G. Li, L. Su, and Q. Huang, “Vehicle detection in UAV traffic video based on convolution neural network,” in Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR), Apr. 2018, pp. 1–6.

[99] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lin, M.-H. Yang, and S. Lyu, “UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking,” Comput. Vis. Image Understand., vol. 193, Apr. 2020, Art. no. 102907.

[100] Q. Zhou and X. Li, “STN-homography: Estimate homography parameters directly,” 2019, arXiv:1906.02539.

[101] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. ICCV, 2017, pp. 2961–2969.

[102] CK Formatted T. Rahman, M. A. L. Siregar, A. Kurniawan, S. Juniastuti, and E. M. Yuniarno, “Vehicle speed calculation from drone video based on deep learning,” in Proc. Int. Conf. Comput. Eng., Netw., Intell. Multimedia (CENIM), Nov. 2020, pp. 229–233.

[103] A. Holla, U. Verma, and R. M. Pai, “Efficient vehicle counting by eliminating identical vehicles in UAV aerial videos,” in Proc. IEEE Int. Conf. Distrib. Comput., VLSI, Elect. Circuits Robot. (DISCOVER), Oct. 2020, pp. 246–251.

[104] R. Collins, X. Zhou, and S. K. Teh, “An open source tracking testbed and evaluation web site,” in Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill., vol. 2, Jan. 2005, p. 35.

[105] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, Jun. 2001, pp. 1–8.

[106] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

[107] M. Fernández-Sanjurjo, B. Bosquet, M. Mucientes, and V. M. Brea, “Real-time visual detection and tracking system for traffic monitoring,” Eng. Appl. Artif. Intell., vol. 85, pp. 410–420, Oct. 2019.

[108] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2117–2125.

[109] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Adv. Neural Inf. Process. Syst. 2016, pp. 379–387.

[110] C. Sun, W. Zhan, J. She, and Y. Zhang, “Object detection from the video taken by drone via convolutional neural networks,” Math. Problems Eng., vol. 2020, Oct. 2020, Art. no. 4013647.

[111] A. G. Howard, M. Zhu, B. Chen, D. Kaltenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

[112] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[113] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C. Bouganis, “DooNet: Efficient convolutional neural network detector for real-time UAV applications,” in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE), Mar. 2018, pp. 967–972.

[114] L. Wen, D. Du, Z. Cai, Z. Lei, S. Yan, and S. Lyu, “UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking,” Comput. Vis. Image Understand., vol. 193, Apr. 2020, Art. no. 102907.

[115] Q. Zhou and X. Li, “STN-homography: Estimate homography parameters directly,” 2019, arXiv:1906.02539.

[116] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. ICCV, 2017, pp. 2961–2969.

[117] CK Formatted T. Rahman, M. A. L. Siregar, A. Kurniawan, S. Juniastuti, and E. M. Yuniarno, “Vehicle speed calculation from drone video based on deep learning,” in Proc. Int. Conf. Comput. Eng., Netw., Intell. Multimedia (CENIM), Nov. 2020, pp. 229–233.

[118] A. Holla, U. Verma, and R. M. Pai, “Efficient vehicle counting by eliminating identical vehicles in UAV aerial videos,” in Proc. IEEE Int. Conf. Distrib. Comput., VLSI, Elect. Circuits Robot. (DISCOVER), Oct. 2020, pp. 246–251.

[119] R. Collins, X. Zhou, and S. K. Teh, “An open source tracking testbed and evaluation web site,” in Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill., vol. 2, Jan. 2005, p. 35.

[120] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1, Jun. 2001, pp. 1–8.
[129] G. A. Mills-Tetley, A. Stentz, and M. B. Dias, “The dynamic Hungarian algorithm for the assignment problem with changing costs,” Robot. Inst., Pittsburgh, PA, USA, Tech. Rep., CMU-RI-TR-07-27, 2007.

[130] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015.

[131] J. Glenn. (Jan. 2021). YOLOv5. Accessed: Nov. 16, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[132] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

[133] Meituan. (Jul. 2022). YOLOv6. Accessed: Jul. 20, 2022. [Online]. Available: https://github.com/meituan/YOLOv6

[134] H. Wu, Y. Chen, N. Wang, and Z.-X. Zhang, “Sequence level semantics aggregation for video object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9217–9225.

[135] Y. Chen, Y. Cao, H. Hu, and L. Wang, “Memory enhanced global-local aggregation for video object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10337–10346.

[136] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, “A review of video object detection: Datasets, metrics and methods,” Appl. Sci., vol. 10, no. 21, p. 7834, Nov. 2020.

[137] M. Suresha, S. Kuppa, and D. S. Raghukumar, “T-CNN: Tubelets with convolutional neural networks for object detection from videos,” IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 10, pp. 2896–2907, Oct. 2018.

[138] H. Wu, Y. Chen, N. Wang, and Z.-X. Zhang, “Sequence level semantics aggregation for video object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9217–9225.

[139] F. Xiao, Y. J. Lee, and P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 941–951.

[141] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLOACT: Real-time instance segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9217–9225.

[142] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLOACT++: Better real-time instance segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 2, pp. 1108–1121, Feb. 2022.

[143] H. Liu, R. A. R. Soto, F. Xiao, and Y. J. Lee, “YolactEdge: Real-time instance segmentation on the edge,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2021, pp. 9579–9585.

[144] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 941–951.

Open Access funding provided by ‘Università degli Studi di Genova’ within the CRUI CARE Agreement

---

**Igor Bisio**: Senior Member, IEEE received the “Laurea” and Ph.D. degrees from the University of Genoa, Italy, in 2002 and 2006, respectively. He is currently an Associate Professor and a member of the Research Staff at the Telecommunication Research Group and, in particular, of the Digital Signal Processing (DSP) Laboratory, University of Genoa. His research interests include IoT signal processing, context and location awareness, adaptive coding, safety and e-health applications, and satellite communication systems. He is an IEEE Communications Society (ComSoc) Member.

**Chiara Garibotto**: (Member, IEEE) was born in Chiavari, Italy in 1985. She received the bachelor’s degree in telecommunication engineering, the master’s degree in multimedia signal processing and telecommunication networks, and the Ph.D. degree in science and technology for electronic and telecommunication engineering from the University of Genoa, in 2012, 2015, and 2019, respectively. She is currently an Assistant Professor with the Digital Signal Processing Laboratory (DSPLab), University of Genoa. Her main research interests include multimedia signal processing, context awareness and intelligent sensing for smart environments, safety and security, and e-health applications.

**Halar Haleem**: (Member, IEEE) received the bachelor’s degree in electronic engineering and the master’s degree in electronic systems engineering from the Mehran University of Engineering and Technology, Pakistan, in 2011 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Digital Signal Processing Laboratory (DSP Lab), University of Genoa, Italy. His main research activities concern surveillance of drones for safety and security applications, deep learning, computer vision, and road traffic monitoring. He was a Student Branch Chapter Advisor of IEEE RA Society, in 2019.

**Fabio Lavagetto**: is currently a Full Professor in telecommunications with the University of Genoa, where he has also been a member of the Board of Directors since 2016. He is also the Head of Research of the Digital Signal Processing (DSP) Laboratory, University of Genoa. He has authored over 100 scientific publications in international journals and conferences. His current research interests include signal processing over Internet of Things, context and location awareness, safety and e-health applications, and satellite communication systems.

**Andrea Sciarrone**: (Senior Member, IEEE) was born in Livorno, Italy, in 1984. He received the bachelor’s degree in telecommunication engineering, the master’s degree (cum laude) in telecommunication engineering, and the Ph.D. degree in ambient intelligence from the University of Genoa, in 2007, 2009, and 2014, respectively. He is currently an Assistant Professor and a member of the Digital Signal Processing (DSP) Laboratory, University of Genoa. His research interests include signal processing over Internet of Things, context and location awareness, and safety and e-health applications. He is currently the Chair of the IEEE ComSoc eHealth Technical Committee.

---

**Open Access funding provided by ‘Università degli Studi di Genova’ within the CRUI CARE Agreement**