IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2022, accepted 24 December 2022, date of publication 5 January 2023,
date of current version 11 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234281

RESEARCH ARTICLE

Smart Traffic Monitoring Through Pyramid Pooling Vehicle Detection and Filter-Based Tracking on Aerial Images

ADNAN AHMED RAFIQUE[1,2], AMAL AL-RASHEED[3], AMEL KSIBI[3], MANEL AYADI[3], AHMAD JALAL[1], KHALED ALNOWAISER[4], HOSSAM MESHREF[5], (Senior Member, IEEE), MOHAMMAD SHORFUZZAMAN[5], (Member, IEEE), MUNKHJARGAL GOCHOO[6,7], (Member, IEEE), AND JEONGMIN PARK[8]

[1]Department of Computer Science, Air University, Islamabad 44000, Pakistan
[2]Department of Computer Sciences and IT, University of Poonch Rawalakot, Rawalakot, Azad Jammu and Kashmir 12350, Pakistan
[3]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[4]Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[5]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia
[6]Department of Computer Science and Software Engineering, United Arab Emirates University, Al Ain, United Arab Emirates
[7]Emirates Center for Mobility Research, United Arab Emirates University, Al Ain, United Arab Emirates
[8]Department of Computer Engineering, Tech University of Korea, Siheung-si, Gyeonggi-do 15073, South Korea

Corresponding authors: Amel Ksibi (amelksibi@pnu.edu.sa) and Jeongmin Park (jmpark@tukorea.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation (NRF) funded by the Ministry of Education, Republic of Korea, under 2021R1F1A1063634; in part by the Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, through Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R235), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia; in part by the Taif University, Taif, Saudi Arabia, through the Taif University Researchers Supporting Project under Grant TURSP-2020/79; and in part by the Emirates Center for Mobility Research (ECMR), United Arab Emirates University, Al Ain, under Grant #12R012.

ABSTRACT Increased traffic density, combined with global population development, has resulted in increasingly congested roads, increased air pollution, and increased accidents. Globally, the overall number of automobiles has expanded dramatically during the last decade. Traffic monitoring in this environment is undoubtedly a significant difficulty in various developing countries. This work introduced a novel vehicle detection and classification system for smart traffic monitoring that uses a convolutional neural network (CNN) to segment aerial imagery. These segmented images are examined to further detect the vehicles by incorporating novel customized pyramid pooling. Then, these detected vehicles are classified into various subcategories. Finally, these vehicles are tracked via Kalman filter (KF) and kernelized filter-based techniques to cope with and manage massive traffic flows with minimal human intervention. During the experimental evaluation, our proposed system illustrated a remarkable vehicle detection rate of 95.78% over the Vehicle Aerial Imagery from a Drone (VAID), 95.18% over the Vehicle Detection in Aerial Imagery (VEDAI), and 93.13% over the German Aerospace Center (DLR) DLR3K datasets, respectively. The proposed system has a variety of applications, including identifying vehicles in traffic, sensing traffic congestion on a road, traffic density at intersections, detecting various types of vehicles, and providing a path for pedestrians.

INDEX TERMS Aerial images, convolutional neural network, correlation filter, traffic monitoring, segmentation, vehicles.

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

I. INTRODUCTION

The technological advancement in remote sensing has increased its popularity and made it more widely available. Recently, several researchers have devoted their efforts to segmentation [1], object recognition [2], [3], [4] scene classification [5], [6], [7], [8] vehicle detection [9], and traffic control systems [10], [11], [12] via aerial and remote sensing (RS) imagery, the list of abbreviations is provided in Table 1. RS and aerial data could significantly boost traffic control, management, efficiency and effectiveness. Traffic management applications include sensing traffic congestion, classifying the different types of vehicles, identifying suspicious vehicles in traffic, and vehicle parking by making vehicle detection a prominent and essential problem in aerial imagery. Although vehicle detection is studied from close-range image data, aerial imagery gives significant information about environments and traffic objects.

The use of a traffic monitoring system is a viable option for reducing traffic jams. The primary function of the traffic monitoring system is to maintain traffic data, such as the number of cars, the kinds of vehicles, and the speed at which they travel. In order to effectively utilize the road network, estimate future transportation requirements, and enhance traveler safety, it conducts traffic analysis using the acquired data. Traffic monitoring systems are usually expensive to create, deploy, and maintain in most countries.

RS and aerial data could significantly boost traffic control and management efficiency and effectiveness. This article focuses on an exciting problem of vehicle detection for traffic monitoring systems using aerial imagery from drones and closed-circuit television (CCTV) cameras. Our work has proposed a novel idea of first segmenting the image, then detecting the vehicle and classifying it into various categories for effective traffic management. Initially, the aerial images are taken as input for semantic segmentation. Then a customized pyramid pooling module (CPPM) is incorporated for vehicle detection in the segmented image. Then, these detected vehicles after classification are grouped into seven categories. Finally, these classified vehicles are tracked by employing two different tracking mechanisms (Kalman filter-based vehicle tracking and kernelized correlation filters-based vehicle tracking). Furthermore, the presented model is validated through the experiments performed over Vehicle Aerial Imagery from Drone (VAID), Vehicle Detection in Aerial Imagery (VEDAI), and German Aerospace Center (DLR3K) datasets. The experiments demonstrated remarkable detection and classification accuracy over other state-of-the-art (SOTA) methods.

The most significant contributions of this work are listed as follows:

- We proposed a novel hybrid framework to detect, classify and track vehicles on roads for efficient management of transportation systems in rural and urban areas.
- A novel vehicle detection via a customized pyramid pooling (CPPM) module is devised for robust traffic monitoring.

TABLE 1. List of abbreviations.

Abbreviations	Definitions
ACF	Aggregated Channel Features
BN	Batch Normalization
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
CPPM	Customized Pyramid Pooling Module
DKF	Distributed Kalman Filter
FT	Fourier Transform
HOG	Histogram of Oriented Gradients
HRPN	Hyper Region Proposal Network
IFM	Input Feature Map
KF	Kalman Filter
LDA	Linear Discriminant Analysis
LiDAR	Light Detection and Ranging
mAP	Mean Accuracy Precision
R-CNN	Region-based CNN
ReLU	Rectified Linear Unit
RFAV	Recurrent-Feature Aware Visualization
RS	Remote sensing
SIFT	Scale-Invariant Feature Transform
SSD	Single-Shot Multibox Detector
SURF	Speeded-up Robust Features
SVM	Support Vector Machine
VAID	Vehicle Aerial Imagery from a Drone
VEDAI	Vehicle Detection in Aerial Imagery

- Two different filter-based tracking approaches: Kalman filter and kernelized filter-based tracking are implemented for vehicle tracking.
- Compared to existing techniques, we have significantly improved the performance metrics including detection rate, precision, recall, F1 Score, and mean accuracy precision for the classification of vehicles.
- The efficiency of the proposed model has been verified over three publicly available datasets in the experimental results, demonstrating outstanding performance.

The remaining part of the paper is organized as follows. The related work is presented in Section II. The proposed methodology and architecture are briefly introduced in Section III, which includes semantic segmentation and vehicle detection using CNN and a CPPM, respectively. Classification of vehicles into seven categories is performed by employing linear discriminant analysis (LDA). Section IV covers the experimental results using aerial and remote sensing data. Section V comprises a discussion of the experiments and results. The conclusion and future work are presented in Section VI.

II. RELATED WORK

Numerous researchers have focused on traffic monitoring systems using machine learning approaches, while others have used deep learning frameworks. Most of the researchers have devoted their efforts to performing vehicle detection and

Reference	Datasets	Methods	Evaluation Metrics	Limitations
S. Javadi et al. [13]	UAV dataset	DarkNet-53, SqueezNet, DenseNet-201	IoU, Recall, Precision and F1-Score	Very costly in terms of training time.
Liu eta al. [14]	Pascal VOC 2012, Pascal Context, SiftFlow	ParseNet with an addition of global features	Mean IoU	The experimental result of this model are not similar across the datasets. Better on SiftFlow but average on Pascal VOC 2012
Tang et al. [15]	Vehicle images dataset	Local Gabor binary pattern and histogram sequence	Detection rate and false rate	
S. Du et al. [16]	UCAS_AOD, VIVID visible, VIVID infrared	Improved YOLOv4, transfer learning	Train loss, Precision, Recall, and F1-Score	Train loss is higher, detection model needs improvements in terms of accuracy.
Huang et al. [17]				
M. Ozturk et al. [18]	COWC dataset	Miniature CNN Architecture	Precision, Recall, F1-Score	Needs extra effort in post-processing. Additionally, other aerial images dataset do not reflect similar accuracies.
Bautista et al. [19]				
Mandal et al. [20]	VEDAI, DLR-3K, DOTA	AVDNet, ConvRes Blocks	Precision, Recall, F1-Score	Only detect vehicles not classify them into various categories.

classification. They incorporated hand-crafted features techniques including scale-invariant feature transform (SIFT), speeded-up robust features (SURF), the histogram of oriented gradients (HOG), and Haar-like features. Once these features are extracted then they applied various machine learning classifiers to detect and classify vehicles in the imagery. These methods are computationally complex and expensive due to their approaches of sliding windows and multi-level search. In the recent past, deep learning-based methods are performing better compared to the previous techniques, particularly for vehicle detection in aerial images and scene understanding tasks. By using convolutional neural networks (CNNs), deep learning-based methods provided superior feature representation than the hand-crafted features and shorter processing times than the sliding window-based methods. CNN-based object detectors are mainly divided into two-step and one-step detectors. Two-step detectors, such as R-CNNs, Fast R-CNN, Faster R-CNN, and Mask R-CNN, use region proposals to complete object location regression and classification processes in two steps. In contrast, one-step detectors, such as YOLOv3 and the single-shot multibox detector (SSD), predict object locations and classes simultaneously in a single network. However, CNN-based methods for vehicle detection in aerial images are limited. Specifically, they perform less satisfactorily in the localization of small objects in a large scene. In addition, training these networks generally demands a high computational cost, and the lack of well-annotated training data adds to the challenge. In this study, we aim to introduce a robust vehicle detection and classification framework that requires limited training data and computational power.

A. LEARNING-BASED VEHICLE DETECTION

For decades, machine learning has been extensively used in computer vision tasks, particularly intelligent traffic management, and monitoring. F. Tang et al. [13] presented a model that considers both the value matrix and spatial-temporal training model while extracting features to predict traffic patterns. They simulated their model and demonstrated a better packet loss rate, average accuracy, and transmission throughput. Liu et al. [14] devised a method to improve the segmentation of the objects and then apply a probabilistic classification model to detect the vehicles correctly. They used aerial images and LiDAR data for the purpose. Tang et al. [15] conducted experiments for vehicle detection on static images by extracting Haar features and then employed an AdaBoost classifier to detect the vehicles in the images. Their approach is practically suitable for various applications of surveillance. Ukani et al. [16] introduced a vehicle detection and classification system that considers video to analyze traffic. They extracted SIFT features for further processing by incorporating the artificial neural network as a classifier as well as a support vector machine (SVM). Their experiments showed better performance when applied SVM. Huang et al. [17] used a combination of background subtraction and a deep belief network to detect the vehicles in a tunnel. It's a challenging problem as different cameras are used in the tunnel. There are also resolution and illumination problems due to reflection on the walls of the tunnel.

B. DEEP LEARNING-BASED VEHICLE DETECTION

Traditionally, traffic monitoring has relied on manual approaches and in-vehicle technologies. However, deep learning-based image processing techniques have surpassed these more traditional ways. In [18], M. Ozturk et al. introduced a framework that uses convolutional neural networks (CNNs) to detect low complexity and high accuracy hybrid vehicles. Morphological operations support this method. They conducted experiments on the COWC dataset and achieved a higher accuracy with fewer parameters compared to the number of parameters used by the other researchers. C. M. Bautista et al. [19] introduced a CNN-based technique that performs the detection and classification of vehicles with the help of low-quality traffic cameras. M. Mandal et al. [20] developed a one-step vehicle detection network (AVDNet) that would be very good at identifying small vehicles. In AVDNet, they added ConvRes residual blocks to handle the small object problem by deeper convolutional layers while extracting features. The larger feature map at output combined with these residual blocks ensures that the important features extracted from small-sized objects are well-represented by the map. They also came up with a way to look at the network's behavior through recurrent-feature aware visualization (RFAV).

In [21], Al-qaness et al. presented a new technique that is used to track vehicles based on video surveillance intelligently. They combined different models to track the vehicles. Initially, they process video by incorporating CNN, and then they use YOLOv3 as an object detection model that is capable to locate the object's position, scale, and category of the object in the image frame. They carried out various experiments to detect objects of different scales including small, medium, and large-scale objects. Moreover, they used average precision, recall, precision, and intersection over union scores to measure the efficiency of the system. Although their proposed system is capable of detecting vehicles on roads and highways. However, there are still some challenges that need to be addressed. For instance, more than 50% of occluded or overlapped objects/vehicles are not correctly detected and tracked. Similarly, nighttime vehicle tracking is a challenge that is not addressed in this study. In [22], Cheng-Jian Lin et al. introduced a three-tier system that is proficient in detecting, counting, and classification of vehicles in different scenarios. They used YOLO for vehicle detection in the first phase. In the second phase, they employed the Kalman filter fused with the Hungarian algorithm to count the vehicles. Finally, a convolutional fuzzy neural network is applied for the classification of vehicles into various categories. Their proposed model is effective to increase the accuracy along with decreasing the parameters. In [23], Peña Cáceres et al. proposed a model to detect the helmet during riding a motorcycle using YOLOv4 algorithms. Their model consists of seven phases including acquiring data, processing it till the completion of the system, and then deployment of the model. They performed various experiments using online platforms. Moreover, they set the ratio to 60:35:5 for training, validation, and testing, respectively while achieving an accuracy of 88.65% detection.

This research aims to contribute to modern world technologies in machine vision. At the same time, the primary purpose of our system is vehicle detection and traffic monitoring to control massive transport. Further, we aim to improve the performance of our system and better results than existing vehicle detection and traffic monitoring systems. Our goal is to try different deep learning techniques to give the best possible vehicle detection accuracy.

III. OUR APPROACH

Initially, the videos containing traffic data are converted to a sequence of frames. These frames are then undergone a segmentation process one by one until the last frame appears. Then, segmented images are analyzed for vehicle detection by employing CPPM. These detected vehicles are also classified into seven different vehicle categories. The detected and classified vehicles are tracked through two different approaches: Kernelazied correlation filter-based vehicle tracking and Kalman filter-based vehicle tracking. Fig. 1 demonstrates the architecture of the proposed model. Moreover, the flow of the proposed model is also provide in Algorithm 1 as follows:

Algorithm 1 Vehicle Detection, Classification, and Tracking Process

```
Input:   Video
Output:  Tracked Vehicles
   a.  vid = VideoReader('video')
   b.  fr = read(vid);
   c.  for frame =1:size(fr)
   d.      re_fr = imresize(fr, 512 × 512)
   e.      seg_obj = Sem_seg(re_fr)
   f.      veh_detect = CPPM (seg_obj)
   g.      veh_class = LDA (veh_detect);
   h.      veh_track1 = ker_filter(veh_class)
   i.      veh_track2 = kal_filer(veh_class)
   j.  compute Acc(veh_track)
   k.      IF Acc(veh_track1) > Acc(veh_track2)
   l.          DISPLAY veh_track1
   m.      ELSE
   n.          DISPLAY veh_track2
```

A. PRE-PROCESSING

To get better results for vehicle detection, tracking, and traffic monitoring, we converted the video into a sequence of images/frames for further processing. Once the frames are extracted from the traffic video, three different types of noise are examined and frames are de-noised by using various filtering techniques. Only the best-suited filter that incorporates the real-time defogging processing of the aerial

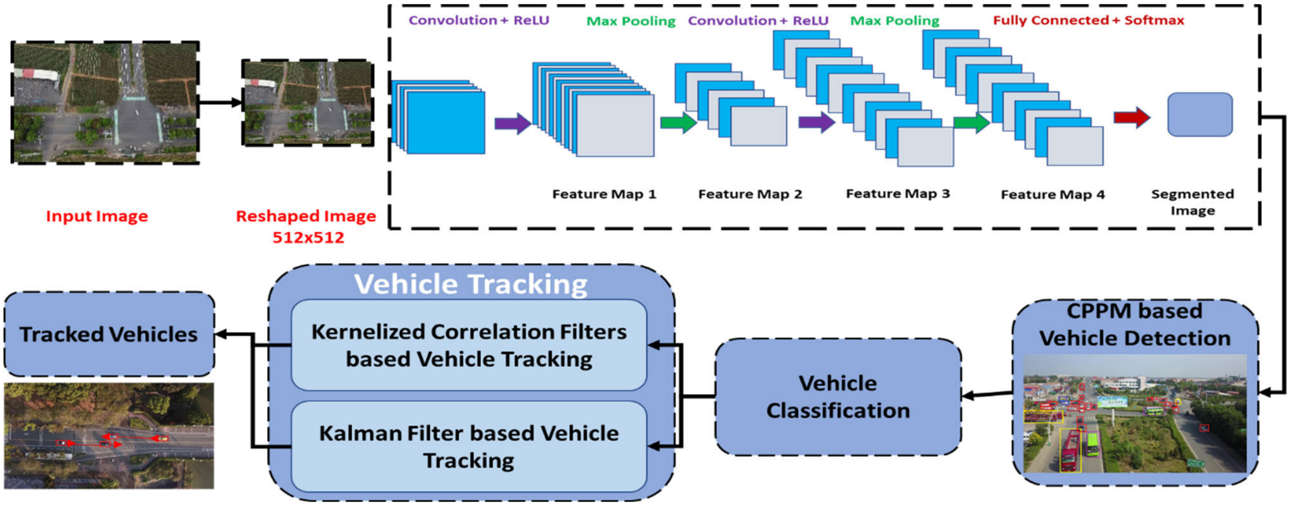

FIGURE 1. System Architecture of Proposed System for Traffic Monitoring.

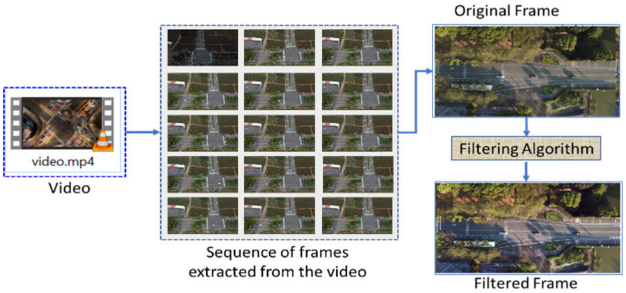

FIGURE 2. Extraction of frames and defogging of frames as a pre-processing step.

images is applied to the respective noise for the best results. The preprocessing step is shown in Fig. 2.

B. CNN-BASED SEMANTIC SEGMENTATION

After the pre-processing phase, image segmentation is performed to separate the vehicles from the other objects and backgrounds. A CNN-based semantic segmentation technique is applied for this purpose. In this phase, a SegNet-based network is described as having two streams. For faster information flow, we used residual blocks with skip connections. Two convolution layers are presented in the residual block, namely conv I and conv II. Layer one comprises 128 filters with a size of 1*1, while the size of filters for other layers is 3*3 with 128 filters. The output produced by the residual block is combined with the output of the second convolutional layer.

In this study, a unique encoder-decoder-based architecture is used. The structure comprises two components: the first component involves five convolution blocks, while the second consists of rectified linear unit (ReLU) and Batch Normalization (BN). By incorporating un-pooling layers in the encoder and decoder, we can restore the resolution to its original state. The encoder and decoder are present in both streams, but at the end of the streams, the combined result of both streams is considered for further processing. A residual block with skip connections is also utilized, as revealed earlier, to send information from each encoder convolution block to its respective encoder-decoder convolution block in both streams. Fig. 3 demonstrates semantic segmentation results over a few examples of the VAID dataset.

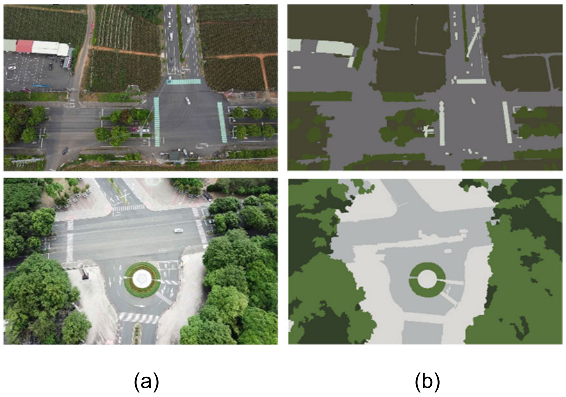

FIGURE 3. CNN-based semantic segmentation of drone-based images from the VAID dataset. (a) original image and (b) segmented image.

In order to get the networks to converge faster, we used pre-trained VGG-16 weights on ImageNet as beginning weights for 50 epochs. The PyTorch framework was used to build the networks. Each convolution block utilizes batch normalization. Network weights are optimized via stochastic gradient descent. The starting learning rate for all decoders and encoders is 0.01 and 0.005, respectively. After 20, 30, and 40 epochs, the learning rate reduces by a factor of 10.

C. VEHICLE DETECTION VIA CUSTOMIZED PYRAMID POOLING MODULE

Local ambiguity can be alleviated by contextual information, as demonstrated in [24]. In VOC2012 [25] and PASCAL-Context [26], ParseNet [14] combined successfully the local features with global pooling to enhance the features set.

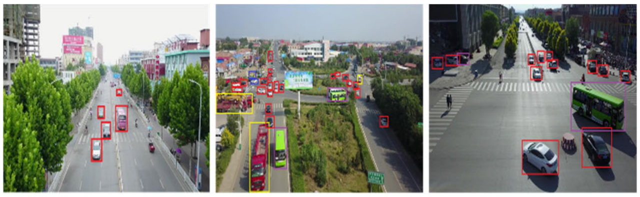

FIGURE 4. Vehicle detection over VAID dataset.

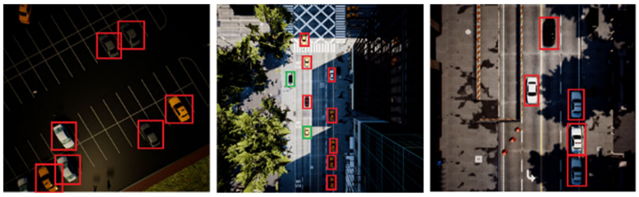

FIGURE 5. Vehicle detection over a few images of the DLR3K complex aerial images dataset.

However, it falls short of what would be required in a more complex scenario. Based on the successful object recognition technique of spatial pyramid pooling, PSPNet [27] integrated various sub-regions to increase inclusive contextual information. There are four sub-region pyramid pooling module scales, including one global pooling layer. The other non-overlapping pooling layers comprised bins with variable sizes. The stride and the kernel size are the same for these non-overlapping layers.

Non-overlapping pooling results in the feature map's spatial size being divided by its kernel size. For this module to work, an input feature map (IFM) must be compatible in terms of a factor of the size of the kernel. Alignment issues could arise as a result of pooling and up-sampling the module. For instance, if the kernel sizes are 40, 20, and 10, then the sum of these kernels is 70, and a multiple of 70 is required for IFM. Unlike the non-overlapping pooling module, the CPPM is more effective. The levels and the kernels are variable-sized and treated as hyperparameters. The first layer is responsible for extracting global features by creating a single bin output. At the same time, the local features are extracted by the other three layers (overlapping pooling layers). The IFM is of fixed size as stride and padding of overlapping pooling layers are kept constant. To reduce the size of the feature map, non-overlapping pooling is performed with a small kernel before applying a CPPM. A max or average pooling operation may be executed. An up-sampling operation with bilinear interpolation is performed to make the feature map compatible. Then, all these three features are fused. The CPPM module is consistent as it uses the IFM of any size as it utilizes the stride of 1 for customized pooling. Fig. 4 and Fig. 5 illustrate the vehicle detection results over some images from the VAID and DLR3K datasets respectively.

D. VEHICLE CLASSIFICATION VIA LINEAR DISCRIMINANT ANALYSIS (LDA)

Linear discriminant analysis [28] is a variant of the Bayesian model. It uses class labels for training purposes as it is a supervised technique. LDA tries to keep intra-class variations low and inter-class variations high. It is employed to classify the detected vehicles into various classes. LDA doesn't need to be scaled since it finds its coefficients based on the difference between the classes. Fig. 6 and 7 show the classification results over the VEDAI and VAID datasets, respectively, where each class is separated, and a total of nine classes are grouped by using the equation as follows:

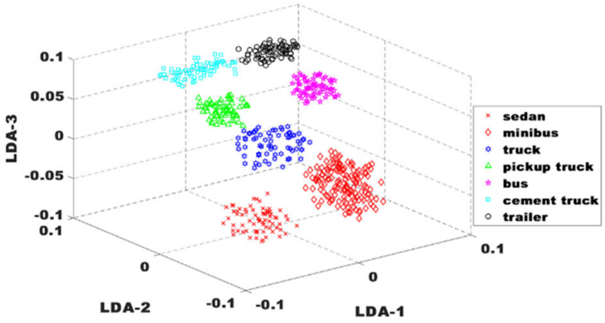

FIGURE 6. Vehicle classification results by applying LDA over the VAID dataset.

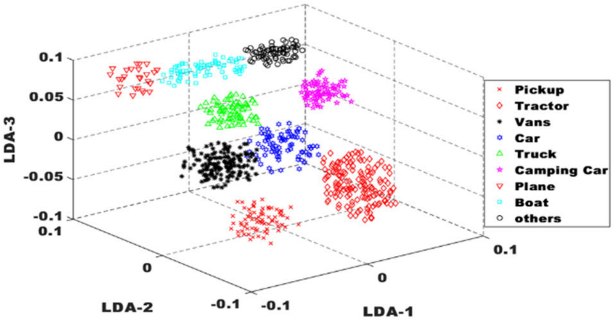

FIGURE 7. Vehicle classification results by applying LDA over the VEDAI dataset with complex aerial images.

$$\Sigma_b = \frac{1}{C}\sum\nolimits_{i=1}^{C}(Meu_i - Meu)(Meu_i - Meu)^T \tag{1}$$

where the mean for all the classes C is denoted by Meu_i, Σ represents the covariance, and Meu is symbolized for the mean of class means.

E. VEHICLE TRACKING VIA KALMAN FILTER METHOD (KER_FILTER)

Kalman filter-based vehicle tracking [29] and its variants [6], [12] are commonly used methods in computer vision tasks and mathematically can be described as follows:

$$X_t = A_t X_t + \omega_t \tag{2}$$

$$Y_t = C_t X_t + v_t \tag{3}$$

where $X_t \in R_n$ is used to represent the state vector, $Y_t \in R_m$ is process noise, $\omega_t \in \mathrm{R}^n$ and $\mathrm{v}_t \in \mathrm{R}^n$ is used to measure noise at step t. Process behavior $A_{t_{mxn}}$ and output matrix $C_{t_{mxn}}$ are the matrices that are commonly used with required dimensions. ω_t and v_t are type of noise.

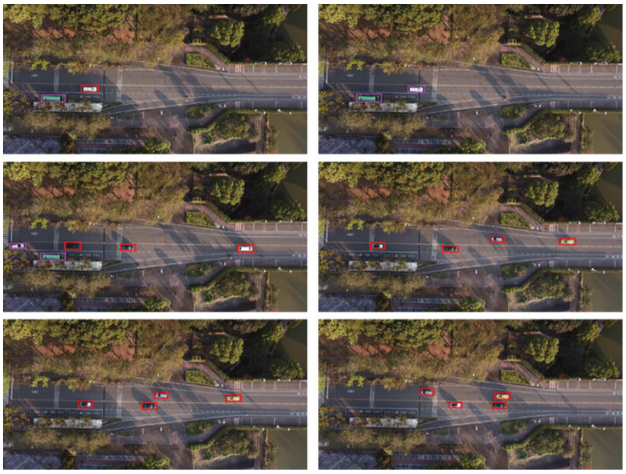

FIGURE 8. **Results of Kalman Filter-based tracking over VAID dataset.**

Kalman filter also uses probabilities in terms of the prior and posterior probability that can be expressed mathematically as follows:

$$\hat{X}_{\bar{t}} = A_{t-1}\hat{X}_{t-1} \tag{4}$$

$$\hat{X}_t = \hat{X}_{\bar{t}} + K_t(Y_t - C_t\hat{X}_{\bar{t}}) \tag{5}$$

Local data collected by each node is relayed to a central server for global estimations, as is the practice in more traditional central approaches. Using KF, all nodes communicate with each other in a decentralized manner. The computation process is heavy and takes a long time. To handle the computation time, alternate methods like distributed Kalman filter (DKF) and diffusion least-mean-square DLMS, are used due to their efficiency based on the information processing mechanism. To DKF, there is no need for a central layer, as every node has the capability that can estimate the system's stale. Fig. 8 illustrates the results of vehicle detection by incorporating the KF tracking.

F. VEHICLE TRACKING VIA KERNELIZED CORRELATION FILTER METHOD (KER_FILTER)

Usually, to identify the target vehicle in the frame, a bounding box around the vehicle is drawn. While considering the correlation filter tracking method [30], highly sampled and circularly shifted image patches are synthesized to build a circular data matrix. This method increases the training sample's capacity without compromising accuracy. The location of the maximum correlation response also aids detection in the successive frames, making it easier to recognize. Given $\mathbf{x} \in \mathbb{R}^{P\times Q\times C}$ where $P \times Q$ denotes the size of the patch with channels C taken from the sample image. All the circulant images $M_{(p,q)}$ with $p < P, q < Q$ are combined to produce the circulant matrix M. Hence, the discrete Fourier transform (FT) is used to compute the eigenvectors of a circulant matrix M:

$$M = F^H Diagonal\,(\hat{m})F \tag{6}$$

where the Hermitian transpose of F is denoted by matrix F^H. The diagonal matrix $\mathcal{F}$(m). *Diagonal*(.) acquired by the corresponding vector and called the FT of x^. The correlation filter **w** and bias b are used to justify the equations:

$$y_i = S\left(\mathbf{w}^\top \mathbf{x}_i + b\right)$$

$$\mathbf{y} = S\left(\mathcal{F}^{-1}\left(\hat{\mathbf{x}}^* \circ \hat{\mathbf{w}}\right) + b\right) \tag{7}$$

Here, all the variants of the original image such as patch $M_{(p,q)}$ are part of the circulant matrix $\boldsymbol{M} = \left[\boldsymbol{M}_{(0,0)}; \boldsymbol{M}_{(0,1)}; \ldots; \boldsymbol{M}_{(P-1,Q-1)}\right]$. Each of new sample $\boldsymbol{M}$ is assigned a unique class label and these class labels are expressed as: $y = [y(0,0), y(0,1), \ldots, y(M-1, N-1)]^T$. while $F^{-1}(\cdot)$ is to represent inverse discrete FT. The difference of the central place $||\mathrm{r}^* - \mathrm{r}_{\mathrm{m,n}}||$ is used to assign the labels of class "y", which is between the region of interest and the image after the circular shift $\mathbf{x}_{(\mathrm{m,n})}$.

$$y_{m,n} = \begin{cases} 1 & \textit{if}\exp\left(-sc \parallel \mathbf{r}_{m,n} - \mathbf{r}^\star ||^{sh}\right) \geq u_o \\ -1 & \textit{if}\exp\left(-sc \parallel \mathbf{r}_{m,n} - \mathbf{r}^\star ||^{sh}\right) \leq l_o \end{cases} \tag{8}$$

where the range of values is represented by l_o and u_o as a minimum and maximum, scale and shape parameters are denoted by sc and sh, respectively. The kernel is represented as the following:

$$\mathbf{w}^\top \psi(\mathbf{x}) = \sum\nolimits_i \alpha_i K\left(\mathbf{x}, \mathbf{x}_i\right) \tag{9}$$

To define $\psi(\mathbf{x})$ which is a non-linear feature mapping, a kernel function $K\left(\mathbf{x}, \mathbf{x}_i\right)$ with the coefficient vector $\alpha = [\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{(\mathrm{M}-1)\times(\mathrm{N}-1)}]^{\mathrm{T}}$ are utilized where K is called positive semi-definite kernel matrix and comprised the elements as follows: $\{\boldsymbol{K_{ij}} = \boldsymbol{K(x_i, x_j\,)}\}_{i\in(0,1,\ldots,\boldsymbol{M}-1)}, j \in (0, 1, \ldots, N-1)$.

To define the kernelized correlation filtering process, eq. (10) can be written by incorporating the properties of circulant matrix K, $\parallel \mathbf{w} \parallel^2 = \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} = \boldsymbol{\alpha}^\top$. Given $\xi = \mathbf{e} + \mathbf{1} - \mathbf{y} \circ \left(\mathcal{F}^{-1}\left(\hat{\mathbf{x}}^* \circ \hat{\mathbf{w}}\right) + b\mathbf{1}\right)$, the linear constraint is represented by $\boldsymbol{e}$, the autocorrelation among the kernels may be computed by $\hat{\mathbf{k}}^{\mathrm{xx}}$ e.g. $\mathbf{k}^{\mathbf{xy}'} = \exp\left(-\left(\parallel \boldsymbol{x} \parallel^2 + \parallel \boldsymbol{y'} \parallel^2 - 2F^{-1}\left(\hat{\boldsymbol{y}}^* \odot \hat{\boldsymbol{x}}'\right)\right)\right)$ (an RBF kernel).

$$\begin{aligned} &\underset{\alpha,b}{min}\, \boldsymbol{\alpha}^\top \mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\mathrm{xx}} \circ \hat{\boldsymbol{\alpha}}\right) \\ &\quad + C\left(\mathbf{y} \circ \left(\mathcal{F}^{-1}\left(\hat{\mathbf{k}}^{\mathrm{xx}} \circ \hat{\boldsymbol{\alpha}}\right) + b\mathbf{1}\right) - \mathbf{1} - \mathbf{e}\right)^2 \\ &\text{s.t. } \mathbf{e} \geq 0. \end{aligned} \tag{10}$$

In this work, before the fusion of kernels, a unique Gaussian kernel to preserve the responses of the filtering, is produced with the help of various features. If we have an *l-th* type of feature vector $x^{(l)}$ having size $M \times N \times D$, then, the training examples of that specific feature vector along their dimensions are computed by the circular shift operation. The estimated response map may be expressed mathematically as follows:

$$f_z^{(l)} = \mathcal{F}^{-1}\left(\hat{\mathrm{k}}^{\mathrm{xz}} \circ \hat{\alpha}_{t-1}^{(l)}\right) + b^{(l)}\mathrm{e} \tag{11}$$

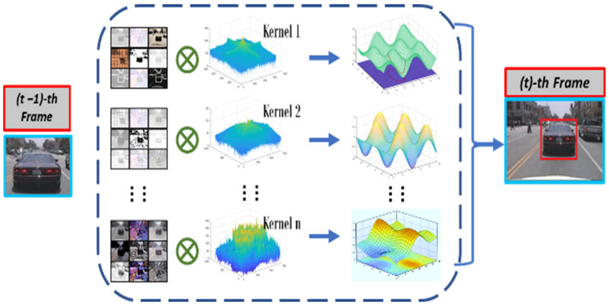

FIGURE 9. Overview of kernelized correlation filter-based tracking.

where the optimal coefficient vector is represented by $\hat{\alpha}_{t-1}^{(l)}$ and the bias is denoted by $b^{(l)}$ at the $(t\ -1)$*-th* frame.

The maximum value of the response map $f_z^{(l)}$ is used to compute the requisite place of the *l-th* feature vector. Multi-kernel correlation responses are integrated into a final distribution map that is dynamically combined using different kernel filters as shown in Fig. 9.

$$f(z) = \sum\nolimits_l f_z^{(l)} * w^{(l)} \tag{12}$$

Scaling parameters can be estimated using variable-scale pyramids, which are able to adjust to variations in appearance. More than one sample is taken from the present target location, and these samples are called ”scale-pool samples” ($\boldsymbol{S} = \{s_1, s_2, s_{3\ldots} s_v\}$). As soon as a new frame becomes available, the highest possible number of v correlation responses can be used to identify both the target's position and its scale at the same time. Normally, we expect the optimal response map to have a sharp peak, but a further decline may cause the response map to be significantly transformed. It is effective to determine the optimal learning rates for the (l) different sorts of feature kernels based on the highest points of respective response maps. We can define the maximum and minimum ability of response as: $P^{(l)} = \frac{R_{max}^{(l)} - R_{min}^{(l)}}{\sigma^{(l)}} \cdot R_{max}^{(l)}$ and $R_{min}^{(l)}$ respectively while $\sigma^{(l)}$ is used to denote the standard deviation. To update the coefficients $\hat{\alpha}_t^{(l)}$ and $b_t^{(l)}$ in the *t-th* frame, a threshold value ($\mathbf{Th} = \frac{\sum_{i=1}^{t-1} P_i^{(t)}}{(t-1)}$) of a classifier PSR (Peak-to-Side lobe Ratio) is utilized.

$$\hat{\boldsymbol{\alpha}}_t^{(l)} = \begin{cases} (1-\eta)\hat{\boldsymbol{\alpha}}_{t-1}^{(l)} + \eta\hat{\boldsymbol{\alpha}}_t^{(l)}, & \mathbf{PSR} < \mathbf{Th} \\ \hat{\boldsymbol{\alpha}}_{t-1}^{(l)}, & \mathbf{PSR} \geq \mathbf{Th} \end{cases}$$
$$b_t^l = \begin{cases} (1-\eta)\hat{b}_{t-1}^{(l)} + \eta\hat{b}_t^{(l)}, & \mathbf{PSR} < \mathbf{Th} \\ \hat{b}_{t-1}^{(l)}, & \mathbf{PSR} \geq \mathbf{Th} \end{cases} \tag{13}$$

where the fusion parameter is called η. Although the original template shape can be preserved to some extent, repetitive pattern filters can also be derived using this method.

IV. EXPERIMENTAL SETUP AND RESULTS

A. DATASETS DESCRIPTION

During our experiments, we have considered three complex aerial imagery datasets including VAID, VEDAI, and DLR3K datasets. The details of these datasets are given as follows:

Algorithm 2 Vehicle Tracking Flow

```
Input: Image frames: $\boldsymbol{I}_t, \boldsymbol{y}$,target center position $p_{t-1}$ and
    scale size $M \times N$ in the $(t\text{-}1)^{th}$ frame.
Output: Current target position $p_{t-1}$ and target scale size
    $M \times N$ in the $(t)^{th}$ frame.
    o. Obtain $\boldsymbol{S}$-scale patches and extract
       corresponding features from $I_t$ at target center
       position $p_{t-1}$.
    p. Calculate multi-kernel response maps by the
       equations (12) and (13).
    q. Update the values of l-th with the help of
       equation (13).
```

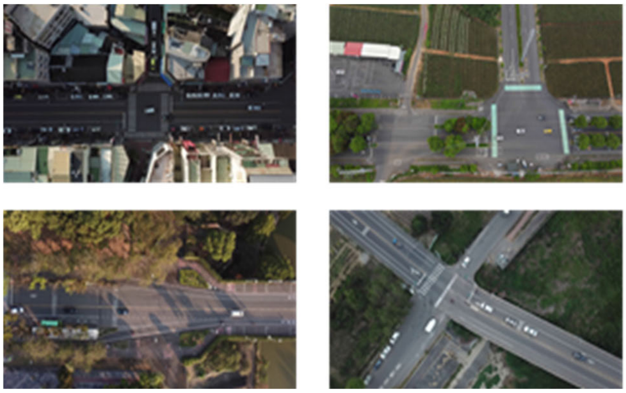

FIGURE 10. Few examples from the VAID dataset.

1) VAID DATASET

The VAID [31] dataset was presented by H.Y. Lin et al. in 2020 for intelligent traffic monitoring via detection and classification of vehicles. The dataset comprised 6000 images of vehicles and was classified into seven different classes such as minibus, cement truck, truck, sedan, pickup truck, bus, and trailer. A drone is used to capture these images in different illumination conditions. The drone is elevated between 90 and 95 meters for consistent images of vehicles. The resolution of images captured at 23.98 frames per second is 2720 × 1530. The images are resized, and pre-processed images' resolution is 1137 × 640. The dataset includes traffic and road conditions for ten places in southern Taiwan. A university campus, a city suburb, and an urban environment are all depicted in the images. Fig. 10 shows the example images from the VAID dataset.

2) VEDAI DATASET

VEDAI [32] is a dataset for vehicle detection in aerial imagery proposed in 2015. The dataset helps researchers find vehicles in aerial images. There are small vehicles in the dataset, and they have various features, like different orientations, lighting, shadow, or occluded objects. A standard protocol is also provided to reproduce and compare the results generated by other researchers. For this dataset, performances

FIGURE 11. Few examples from the VEDAI dataset.

FIGURE 12. Few examples from the DLR3K dataset.

of some baseline algorithms are also given. Fig. 11 illustrates some images from the VEDAI dataset.

3) DLR-3K DATASET

DLR-3K dataset [33] is a collection of various aerial scenes of vehicles from urban as well as some residential areas. The dataset is also known as DLR Munich vehicle detection dataset and comprised 20 images of high resolution (5616 × 3744) with vehicle types including ‘‘car’’ and ‘‘truck’’. The number of images having the ‘‘car’’ class is more than that of the other type of vehicles. To train the model, original images are divided into nine parts (3 × 3) which results in a total of 180 images. A few example images of the DLR3K dataset are shown in Fig. 12.

B. IMPLEMENTATION DETAILS

To implement the system, we set an environment by using python 3.7. The vehicle detection results are based on the CPPM and the detected vehicles are marked with bounding boxes around them. The performance of detection depends upon the minimum threshold that is set to detect an object and intersection over the union score. The object and class confidence values are computed, as given below:

$$IoU = \frac{\text{Area of Overlapping}}{Area\ of\ Union} \tag{14}$$

TABLE 2. Parameters used during the training of the model.

Parameter Name	Value/Range
Mini-batch size	04
Weight update ratio	0.0005
Value of Momentum	0.9
Rate of Learning (initial)	0.001
Input layer size	608x608

1) TRAINING CONFIGURATION

A system with a GeForce RTX 3080 Ti GPU is used to train the model. To determine the input layer size and other parameters, parameter sensitivity analysis is performed that authenticates the computational performance as well as the accuracy of the model. The sum of square errors from the final layer of the network is used to compute the training loss. The details of the parameters used during the training process are described in Table 2.

2) MODEL TRAINING

For the model training, we considered train and test sets with a ratio of 80:20 for VEDAI and VAID respectively. On the other hand, a 70:30 ratio was applied over the DLR-3K data set for the train and test respectively. The proposed model is used to train over each dataset and during the training no pre-trained weights are used. The proposed model over VEDAI, VAID, and DLR-3K datasets executed 20k iterations during training. The learning rate is changed after each 5K iterations by a factor of 100. Multiple bounding boxes are generated for each object. The object with the highest score of IoU is selected in the proposed model on the basis of the specified threshold.

C. RESULTS AND ANALYSIS

1) QUANTITATIVE ANALYSIS

In this section, we conducted experiments to record the detection and classification accuracy of the proposed model over benchmark datasets in order to ensure its validity compared to other existing methods.

a: DETECTION ACCURACIES

The evaluation of the proposed model is conducted over three benchmark datasets: VEDAI, DLR-3K, and VAID. We computed the different metrics including mean accuracy precision (mAP), specificity, recall, precision, and F1 Score. The detailed analysis of the metrics is recorded in Tables 3 and 4 over VAID and VEDAI datasets respectively. In order to certify fairness, a similar set of unseen samples from the test data is used to evaluate the proposed model. The results showed remarkable performance over the existing state-of-the-art techniques.

TABLE 3. The overall accuracy, precision, recall, F1 score, specificity, and computational time for vehicle detection results were obtained using customized pyramid pooling over the VAID dataset.

Vehicle Class	Detection Accuracy	Precision	Recall	F1 Score	Specificity	Average Computational time (seconds/class)
Sedan	97.27	0.9646	0.9433	0.9538	0.923	181
Minibus	96.66	0.9658	0.8719	0.9165	0.905	201
Truck	98.01	0.9602	0.8673	0.9114	0.911	185
Pickup Truck	94.75	0.9588	0.9789	0.9687	0.901	217
Bus	98.57	0.9643	0.8529	0.9052	0.856	213
Cement Truck	91.29	0.9418	0.8845	0.9123	0.891	225
Trailer	93.89	0.8955	0.7969	0.8433	0.843	194
Mean	**95.78**	**0.9501**	**0.8851**	**0.9159**	**0.8926**	**202.29**

TABLE 4. The overall precision, recall, F1 Score, specificity, accuracy, and computational time for vehicle detection results were obtained using a customized pyramid pooling technique over the VEDAI dataset.

Vehicle Class	Detection Accuracy	Precision	Recall	F1 Score	Specificity	Average Computational Time (seconds/class)
Pickup	95.57	0.9154	0.8911	0.9031	0.915	239
Tractor	93.74	0.9224	0.8415	0.8801	0.897	213
Vans	95.66	0.9517	0.8801	0.9145	0.908	228
Car	96.89	0.9432	0.8752	0.9079	0.911	207
Truck	97.91	0.9115	0.8967	0.904	0.884	199
Camping Car	91.22	0.9146	0.8477	0.8799	0.829	221
Plane	97.56	0.9398	0.8623	0.8994	0.817	195
Boat	95.85	0.9345	0.9175	0.9259	0.873	216
Others	92.15	0.9095	0.8551	0.8815	0.855	203
Mean	**95.17**	**0.9269**	**0.8741**	**0.8996**	**0.8765**	**213.44**

b: CLASSIFICATION ACCURACIES

In this section, experiments are executed to validate the significance of our proposed system. To present the results, we computed the confusion matrix for vehicle classification over the VAID and VEDAI datasets as shown in Tables 5 and 6. It is evident from Table 5 that the proposed model has significant results with an average classification accuracy of 96.71% over the VAID dataset. Moreover, *sedan* and *trailer* classes have the highest accuracy, while *cement truck* has the lowest accuracy. Similarly, Table 6 depicts that the *car* class achieved the highest classification accuracy while the *camping car* lies at the bottom-most in the classification accuracy list.

2) QUALITATIVE ANALYSIS

We examined the results of our proposed model qualitatively in a variety of challenging circumstances. The semantic segmentation and detection of vehicles in three benchmark datasets are performed with higher mean accuracy and classification results are tremendous by applying the proposed model. Additionally, the proposed system is smart enough to identify partially occluded vehicles as shown in Figure 13 (a) where some partially occluded vehicles under the shade are detected by the system and while in Figure 13 (b) some vehicles are occluded under the trees yet the system is able to detect these vehicles. Our proposed system is also helpful in the robust detection of vehicles with different shapes and

TABLE 5. Confusion Matrix of the classification accuracies over various vehicle classes of VAID dataset.

Vehicle Class	Sedan	Minibus	Truck	Pickup Truck	Bus	Cement Truck	Trailer
Sedan	**0.99**	0	0	0.01	0	0	0
Minibus	0	**0.94**	0	0	0.06	0	0
Truck	0	0	**0.98**	0.02	0	0	0
Pickup Truck	0	0	0	**0.97**	0.03	0	0
Bus	0	0.02	0	0	**0.98**	0	0
Cement Truck	0	0	0.08	0	0	**0.92**	0
Trailer	0	0	0	0	0	0.01	**0.99**
Mean = 96.71%							

TABLE 6. Confusion Matrix for the classification accuracies over various vehicle classes of the VEDAI dataset.

Vehicle Class	Pickup	Tractor	Vans	Car	Truck	Camping Car	Plane	Boat	Others
Pickup	**0.97**	0	0	0	0.03	0	0	0	0
Tractor	0	**0.96**	0	0	0	0.04	0	0	0
Vans	0.02	0	**0.94**	0	0.04	0	0	0	0
Car	0	0	0.02	**0.98**	0	0	0	0	0
Truck	0.02	0	0.04	0	**0.94**	0	0	0	0
Camping Car	0	0	0.05	0	0	**0.92**	0	0	0.03
Plane	0	0	0	0	0	0	**0.95**	0	0.05
Boat	0	0	0	0	0	0.02	0	**0.95**	0.03
Others	0	0	0	0	0	0	0	0.04	**0.96**
Mean Accuracy = 95.22%									

orientations Furthermore, the system is also capable of detecting vehicles sheltered due to the shadow of other objects.

Although there are some failure cases where the occlusion is more than 50% of the object in terms of pixels as shown in Figure 14. Where most part of the vehicle is occluded under a tree, the system is unable to detect it as a vehicle. However, the overall qualitative results validate the performance of our system in a diversity of challenging environments.

3) COMPLEXITY ANALYSIS

The computation and space complexity of the proposed system is computed in terms of the number of parameters and model size. The complexity comparison with current SOTA methods is illustrated in Table 7. It is evident from the statistics that the proposed model has fewer parameters when compared with existing techniques including YOLO, RetinaNet, and Faster R-CNN. Moreover, the proposed model requires less memory space as compared to YOLO and other deep networks like Faster R-CNN, or RetinaNet. As a result, the proposed method is more efficient (in terms of computation and memory space) than the current SOTA methods.

TABLE 7. Comparative analysis of the proposed model with other existing techniques in terms of computational and space complexity.

Method	No. of parameters (in millions)	Model Size
Yolo v2	67	255MB
Yolo v3	61	235MB
Faster R-CNN	59	253MB
RetinaNet	36	146MB
Proposed Mode	**29**	**95MB**

4) COMPARATIVE ANALYSIS

We evaluated our model and compared it with the existing techniques available in the literature and considered it to be the SOTA technique. Tables 8, demonstrates the comparison

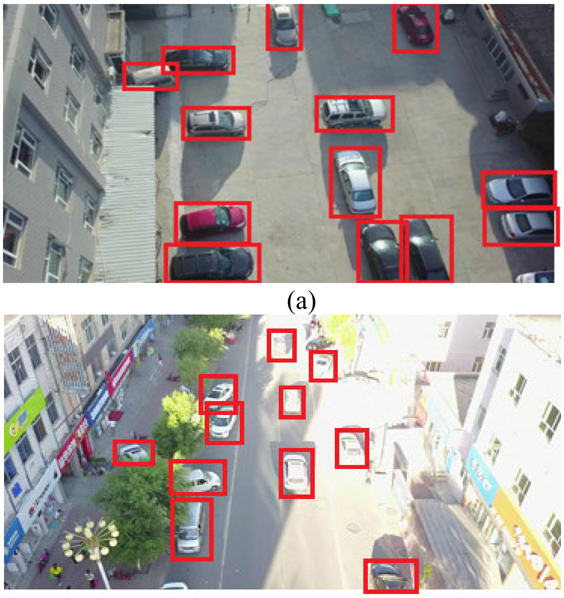

FIGURE 13. Vehicle detection in case of occlusion (a) occluded under shades of building, (b) occluded under the trees. failure case where a vehicle is occluded under the tree.

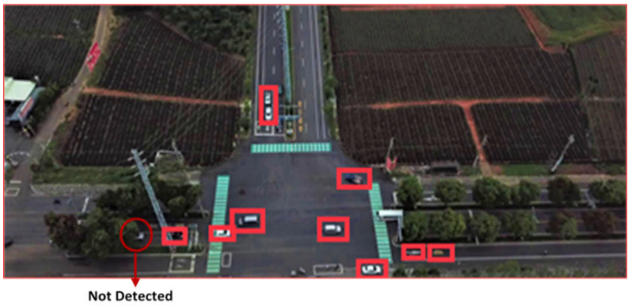

FIGURE 14. Failure case where a vehicle is occluded under the tree.

of classification accuracies between the proposed model and SOTA techniques on VEDAI and VAID datasets while Table 9 illustrates the comparison of detection accuracies over the DLR3K dataset.

V. DISCUSSION

The proposed traffic monitoring system was designed to manage traffic over high-resolution aerial imagery. In this study, we developed a framework that uses CNN-based semantic segmentation to effectively segment out the objects more specifically the vehicles in the aerial images. These segmented objects are further examined for vehicle detection through the proposed CPPM. Then the detected vehicles are categorized into different classes. Additionally, all the categorized vehicles are tracked by employing Kalman filter and kernelized filter-based tracking techniques. Both techniques produced good results yet kernelized filter-based tracking super pass the earlier one.

TABLE 8. Vehicle Classification Accuracies Comparison with other SOTA Methods over the VEDAI and VAID Datasets.

Methods	VEDAI Dataset	VAID Dataset
	Accuracy (%)	Accuracy (%)
B. Wang et al. [34]	93.96	--
M. Mandal et al. [20]	51.95	--
J. O. Terrail et al. [35]	83.50	--
B. Wang et al. [36]	91.27	--
Y. H. Lin et al. [31]	--	89.3
Proposed Model	**92.22**	**96.71**

TABLE 9. Comparison of vehicle detection results with other SOTA techniques over the DLR3K dataset.

Methods	Accuracy (%)
Darknet 19 [37]	90.51
AVDNet [20]	56.24
Zhong J. et al. [38]	73.70
Darknet 53 [37]	84.25
Proposed Model	**93.13**

Primarily, high-resolution aerial images are very critical, particularly when dealing with vehicle detection. Therefore, an effective mechanism of CNN-based semantic segmentation was incorporated to achieve significant results for segmented regions from the complex high-resolution scene images. Once the aerial images are segmented, they are analyzed to detect different vehicles. The detection phase is the most important part of the system where a novel CPPM technique is devised that effectively enhance the efficiency of the overall system. There is a significant increase in accuracy, precision, and recall in both detection and classification as a result of CPPM. Moreover, the proposed CPPM technique supplements the effective tracking of the vehicles once detected effectively.

We applied various techniques including hyper region proposal network (HRPN), Faster R-CNN, aggregated channel features (ACF) detector, and CPPM to authenticate the validity of our proposed detection mechanism. For this purpose, the same benchmarks i.e. VAID, VEDAI, and DLR-3k are considered for vehicle detection. The average detection accuracies of these techniques are shown in Fig. 15. It is evident from Figure 13 that CPPM outperforms the other techniques in terms of detection accuracy over benchmark datasets. It is observed that ACF has the lowest performance compared to Faster R-CNN, HRPN, and CPPM (our proposed) vehicle

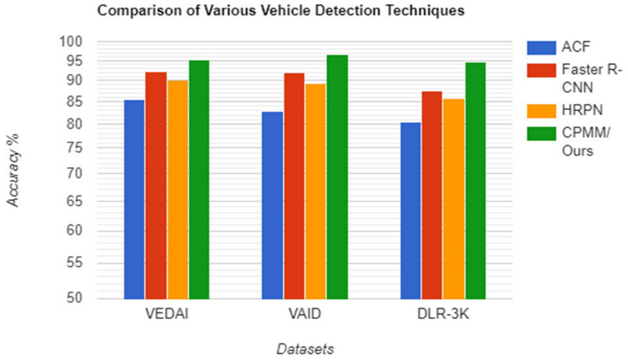

FIGURE 15. Accuracy results of various techniques for vehicle detection.

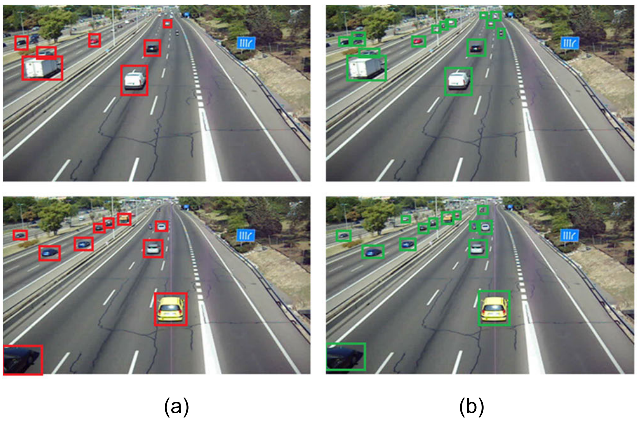

FIGURE 16. Tracking results of both the techniques (a) Kalman filter-based vehicle tracking, (b) kernelized filter-based vehicle tracking.

detection techniques. Moreover, our proposed CPPM has remarkably performed over all the considered datasets.

Moreover, once the vehicles are detected, they are further investigated for classification purposes. These classified vehicles are then tracked via two unique techniques i.e. Kalman filter-based tracking and kernelized filter-based tracking. The accuracy of both the tracking algorithms is compared after tracking. The algorithm having better accuracy is adopted for final vehicle tracking. In most cases, the kernelized filter-based tracking algorithm has shown better tracking results compared to Kalman filter-based tracking. Hence, kernelized filter-based tracking is adopted for the tracking. The results of both Kalman and Kernelized filter based tracking are shown in Figure 16.

VI. CONCLUSION

This paper presents a framework for vehicle detection over aerial images from drones. The proposed model potentially deals with intelligent traffic monitoring, traffic management, and smart surveillance systems. The novel traffic monitoring system enhanced the efficiency of vehicle detection based on the proposed customized pyramid pooling module. The initial module efficiently segments the images before applying the novel customized pyramid pooling module to detect various vehicles in the aerial images. These vehicles are then classified into different categories via linear discriminant analysis. Finally, these classified vehicles are tracked via Kalman filter and kernelized filter-based tracking. The effectiveness of our methodology is not only validated on the VAID, VEDAI, and DRL3K datasets, however, a comparison with other SOTA methods also demonstrated the significance of the proposed method by the experimental results.

The datasets considered for experiments are complex as well as dynamic and diverse backgrounds with different types of vehicles. These scene images are collected from various locations including rural and urban areas. Due to the dynamic scenes with messy information about vehicles along with cluttered backgrounds, our proposed detection module (CPPM) over different datasets responded differently. We faced difficulties under conditions like partially or fully occluded, covered under trees or shades of buildings and similar objects. In future work, we are planning to improve the effectiveness of vehicle tracking using an end-to-end deep learning method for overall traffic monitoring based on vehicle detection and tracking for surveillance.

REFERENCES

[1] T. Zhang, X. Zhang, P. Zhu, X. Tang, C. Li, L. Jiao, and H. Zhou, "Semantic attention and scale complementary network for instance segmentation in remote sensing images," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10999–11013, Oct. 2022.

[2] Y. Zhong, X. Han, and L. Zhang, "Multi-class geospatial object detection based on a position-sensitive balancing framework for high spatial resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 281–294, Apr. 2018.

[3] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2337–2348, Apr. 2018.

[4] X. Li, B. Liu, G. Zheng, Y. Ren, S. Zhang, Y. Liu, Le Gao, Y. Liu, B. Zhang, and F. Wang, "Deep-learning-based information mining from ocean remote-sensing imagery," *Nat. Sci. Rev.*, vol. 7, no. 10, pp. 1584–1605, 2020.

[5] G. S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Apr. 2017.

[6] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1–13, Sep. 2019.

[7] Y. Li, Y. Zhang, and Z. Zhu, "Error-tolerant deep learning for remote sensing image scene classification," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 1756–1768, Apr. 2021.

[8] X. Tang, F. Meng, X. Zhang, Y.-M. Cheung, J. Ma, F. Liu, and L. Jiao, "Hyperspectral image classification based on 3-D octave convolution with spatial–spectral attention network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 1–25, Mar. 2021.

[9] J. Shen, N. Liu, and H. Sun, "Vehicle detection in aerial images based on lightweight deep convolutional network," *IET Image Process.*, vol. 15, no. 2, pp. 479–491, Feb. 2021.

[10] 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.

[11] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, "Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 70–81, Summer 2019.

[12] Q. N. Naveed, H. Alqahtani, R. U. Khan, S. Almakdi, M. Alshehri, and M. A. A. Rasheed, "An intelligent traffic surveillance system using integrated wireless sensor network and improved phase timing optimization," *Sensors*, vol. 22, no. 9, p. 3333, Apr. 2022.

[13] F. Tang, B. Mao, Z. M. Fadlullah, J. Liu, and N. Kato, ''ST-DeLTA: A novel spatial–temporal value network aided deep learning based intelligent network traffic control system,'' *IEEE Trans. Sustain. Comput.*, vol. 5, no. 4, pp. 568–580, Oct. 2020.

[14] W. Liu, A. Rabinovich, and A. C. Berg, ''ParseNet: Looking wider to see better,'' 2015, *arXiv:1506.04579*.

[15] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, ''Vehicle detection and recognition for intelligent traffic surveillance system,'' *Multimedia Tools Appl.*, vol. 76, no. 4, pp. 5817–5832, 2017.

[16] V. Ukani, S. Garg, C. Patel, and H. Tank, ''Efficient vehicle detection and classification for traffic surveillance system,'' in *Proc. Int. Conf. Adv. Comput. Data Sci.* Singapore: Springer, 2016, pp. 495–503.

[17] B. J. Huang, J. W. Hsieh, and C. M. Tsai, ''Vehicle detection in Hsuehshan tunnel using background subtraction and deep belief network,'' in *Proc. Asian Conf. Intell. Inf. Database Syst.* Cham, Switzerland: Springer, 2017, pp. 217–226.

[18] M. Ozturk and E. Cavus, ''Vehicle detection in aerial imaginary using a miniature CNN architecture,'' in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2021, pp. 1–6.

[19] C. M. Bautista, C. A. Dy, M. I. Manalac, R. A. Orbe, and M. Cordel, ''Convolutional neural network for vehicle detection in low resolution traffic videos,'' in *Proc. IEEE Region 10 Symp. (TENSYMP)*, May 2016, pp. 277–281.

[20] M. Mandal, M. Shah, P. Meena, S. Devi, and S. K. Vipparthi, ''AVDNet: A small-sized vehicle detection network for aerial visual data,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 494–498, Mar. 2020.

[21] M. A. A. Al-Qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, ''An improved YOLO-based road traffic monitoring system,'' *Computing*, vol. 103, no. 2, pp. 211–230, Feb. 2021.

[22] C.-J. Lin and J.-Y. Jhang, ''Intelligent traffic-monitoring system based on Yolo and convolutional fuzzy neural networks,'' *IEEE Access*, vol. 10, pp. 14120–14133, 2022.

[23] P. Cáceres, A. M. More-More, F. J. Yáñez-Palacios, T. Samaniego-Cobo, and J. Vargas-Vargas, ''Detection of motorcyclists without a safety helmet through YOLO: Support for road safety,'' in *Proc. Int. Conf. Technol. Innov.* Cham, Switzerland: Springer, 2022, pp. 107–122.

[24] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, ''CNN based semantic segmentation for urban traffic scenes using fisheye camera,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 231–236.

[25] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, ''The PASCAL visual object classes challenge: A retrospective,'' *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.

[26] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, ''The role of context for object detection and semantic segmentation in the wild,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 891–898.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, ''Pyramid scene parsing network,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[28] H. Wang, C. Ding, and H. Huang, ''Multi-label linear discriminant analysis,'' in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 126–139.

[29] M. B. Khalkhali, A. Vahedian, and H. S. Yazdi, ''Multi-target state estimation using interactive Kalman filter for multi-vehicle tracking,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1131–1144, Mar. 2020.

[30] X. An, Q. Liang, and N. Sun, ''Multi-kernel support correlation filters with temporal filtering constraint for object tracking,'' *Multimedia Tools Appl.*, vol. 80, no. 9, pp. 14041–14073, Apr. 2021.

[31] H.-Y. Lin, K.-C. Tu, and C.-Y. Li, ''VAID: An aerial image dataset for vehicle detection and classification,'' *IEEE Access*, vol. 8, pp. 212209–212219, 2020.

[32] S. Razakarivony and F. Jurie, ''Vehicle detection in aerial imagery: A small target detection benchmark,'' *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, Jan. 2016.

[33] K. Liu and G. Mattyus, ''Fast multiclass vehicle detection on aerial images,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[34] B. Wang and B. Xu, ''A feature fusion deep-projection convolution neural network for vehicle detection in aerial images,'' *PLoS ONE*, vol. 16, no. 5, May 2021, Art. no. e0250782.

[35] J. O. du Terrail and F. Jurie, ''Faster RER-CNN: Application to the detection of vehicles in aerial images,'' 2018, *arXiv:1809.07628*.

[36] B. Wang and Y. Gu, ''An improved FBPN-based detection network for vehicles in aerial images,'' *Sensors*, vol. 20, no. 17, p. 4709, 2020.

[37] Z. Wang, D. Liu, Y. Lei, X. Niu, S. Wang, and L. Shi, ''Small target detection based on bird's visual information processing mechanism,'' *Multimedia Tools Appl.*, vol. 79, nos. 31–32, pp. 22083–22105, Aug. 2020.

[38] J. Zhong, T. Lei, and G. Yao, ''Robust vehicle detection in aerial images based on cascaded convolutional neural networks,'' *Sensors*, vol. 17, no. 12, p. 2720, Nov. 2017.

ADNAN AHMED RAFIQUE is currently pursuing the Ph.D. degree with the Department of Computer Science, Air University, Islamabad, Pakistan. He is also working as a Lecturer of computer science with the University of Poonch Rawalakot, Azad Jammu and Kashmir, Pakistan. His research interests include artificial intelligence, machine learning, and computer vision.

AMAL AL-RASHEED received the Ph.D. degree in information systems from King Saud University, in 2017. She is currently an Associate Professor with the Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), Riyadh, Saudi Arabia. She has been involved in many projects related to learning technologies, cyber security, and virtual reality. Her contributions in research projects in academia led to the publication of papers in many journals and conferences. Her research interests include education, knowledge management, data mining, data analytics, cyber security, and natural language processing. In 2017, she was awarded the Research Excellence Award, by PNU, for her publications during performing Ph.D. degree.

AMEL KSIBI received the B.S., M.S., and Ph.D. degrees in computer engineering from the National School of Engineering of Sfax (ENIS), Sfax University, Tunisia, in 2008, 2010, and 2014, respectively. She spent three years at ENIS as a Teaching Assistant, before joining the Higher Institute of Computer Science and Multimedia Gabes (ISIMG) as a Permanent Lecturer, in 2013. She joined the Computer Science Department, Umm Qura University (UQU), as an Assistant Professor, in 2014. After, she joined Princess Nourah bint Abdulrahman University, in 2018, where she is currently an Assistant Professor with the Department of Information Systems, College of Computer Sciences and Information. Her research interests include computer vision, image processing, deep learning, information retrieval, lifelogging and wellbeing, smart education, smart agriculture, and sustainable environment.

MANEL AYADI received the Ph.D. degree in computer science from Le Havre University, France, and the M.Sc. degree in computer science from the University of Sfax, Tunisia. In 2018, she joined Princess Nourah bint Abdulrahman University, where she is currently an Assistant Professor with the Department of Information Systems, College of Computer Sciences and Information (CCIS-IS). Her research interests include information systems engineering, image processing, machine learning, e-learning, and the Internet of Thing. She was a member of the Scientific Committee and/or an organization committee of a number of international conferences.

AHMAD JALAL received the Ph.D. degree from the Department of Biomedical Engineering, Kyung Hee University, Republic of Korea. He is currently an Associate Professor with the Department of Computer Science and Engineering, Air University, Pakistan. He was a Postdoctoral Research Fellow at POSTECH. His research interests include multimedia contents and artificial intelligence.

KHALED ALNOWAISER received the Ph.D. degree in computer science from Glasgow University, U.K. He is currently an Assistant Professor with the Department of Computer Engineering, Prince Sattam bin Abdulaziz University, Saudi Arabia. His research interests include computer vision, optimization techniques, and performance enhancement.

HOSSAM MESHREF (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Virginia Tech University, Blacksburg, VA, USA, in 2002. He is currently an Associate Professor with the Department of Computer Science, Taif University, Taif, Saudi Arabia. His main research interests include machine learning, data science, and multimedia security. He is an active member of several IEEE societies.

MOHAMMAD SHORFUZZAMAN (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Manitoba, Winnipeg, MB, Canada, in 2005 and 2012, respectively. He is currently an Associate Professor with the Department of Computer Science, College of Computers and Information Technology (CCIT), Taif University, Taif, Saudi Arabia. He is also a member of the Big Data Analytics and Applications (BDAAG) Research Group, CCIT. His current research interests include applied artificial intelligence in the areas of computer vision, natural language processing, big data, and cloud computing.

MUNKHJARGAL GOCHOO (Member, IEEE) was born in Ulaanbaatar, Mongolia, in 1984. He received the B.S. and M.S. degrees in electronics engineering from the Mongolian University of Science and Technology, in 2004 and 2005, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Taipei University of Technology, Taiwan. He was with the Department of Electronics, Mongolian University of Science and Technology, as a Lecturer, from 2005 to 2011. His main research interests include telecare, eldercare, the Internet of Things, machine learning, and deep learning classification algorithms.

JEONGMIN PARK received the Ph.D. degree from the College of Information and Communication Engineering, Sungkyunkwan University, in 2009. He is currently an Associate Professor with the Department of Computer Engineering, Tech University of Korea, South Korea. Before joining Tech University of Korea, in 2014, he was a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI) and a Research Professor at Sungkyunkwan University, South Korea. His research interests include high-reliable autonomic computing mechanism and human-oriented interaction systems.