IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2020, accepted December 30, 2020, date of publication January 6, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049741

Vehicle Detection in Aerial Images Based on 3D Depth Maps and Deep Neural Networks

SALEH JAVADI[1], (Student Member, IEEE), MATTIAS DAHL[2], (Member, IEEE), AND MATS I. PETTERSSON[2], (Member, IEEE)

[1]Department of Mathematics and Natural Sciences, Blekinge Institute of Technology (BTH), 37435 Karlshamn, Sweden
[2]Department of Mathematics and Natural Sciences, Blekinge Institute of Technology (BTH), 37179 Karlskrona, Sweden

Corresponding author: Saleh Javadi (saleh.javadi@bth.se)

This work was supported by the Municipality of Karlshamn, Sweden.

ABSTRACT Object detection in aerial images, particularly of vehicles, is highly important in remote sensing applications including traffic management, urban planning, parking space utilization, surveillance, and search and rescue. In this article, we investigate the ability of three-dimensional (3D) feature maps to improve the performance of deep neural network (DNN) for vehicle detection. First, we propose a DNN based on YOLOv3 with various base networks, including DarkNet-53, SqueezeNet, MobileNet-v2, and DenseNet-201. We assessed the base networks and their performance in combination with YOLOv3 on efficiency, processing time, and the memory that each architecture required. In the second part, 3D depth maps were generated using pairs of aerial images and their parallax displacement. Next, a fully connected neural network (fcNN) was trained on 3D feature maps of trucks, semi-trailers and trailers. A cascade of these networks was then proposed to detect vehicles in aerial images. Upon the DNN detecting a region, coordinates and confidence levels were used to extract the corresponding 3D features. The fcNN used 3D features as the input to improve the DNN performance. The data set used in this work was acquired from numerous flights of an unmanned aerial vehicle (UAV) across two industrial harbors over two years. The experimental results show that 3D features improved the precision of DNNs from 88.23 % to 96.43 % and from 97.10 % to 100 % when using DNN confidence thresholds of 0.01 and 0.05, respectively. Accordingly, the proposed system was able to successfully remove 72.22 % to 100 % of false positives from the DNN outputs. These results indicate the importance of 3D features utilization to improve object detection in aerial images for future research.

INDEX TERMS Convolutional neural networks, 3D depth maps, object detection, aerial images.

I. INTRODUCTION

Vehicle detection in aerial images is a key factor in understanding complex transportation systems [1]. Traffic management, urban planning, parking space utilization, surveillance, and search and rescue are among a wide variety of applications that require vehicle detection [2]–[6].

Development and availability of unmanned aerial vehicle (UAV) platforms have made aerial imaging relatively inexpensive, convenient, and suitable for real-time performance applications [7], [8]. In addition, high-resolution UAV images captured from low altitudes can cover a large area with little cloud interference [1]. Therefore, UAV photography is a useful supplement to satellites and aircraft for remote sensing applications [9].

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

However, detecting vehicles in aerial images both accurately and quickly is challenging. As aerial images are taken from altitude with a top-down view, vehicles appear relatively small, and a single image may contain many vehicles [10]. Moreover, other objects, shadows, and various patterns, such as road markings, can appear similar to vehicles [3]. Furthermore, in comparison to video analysis from a stationary camera, as UAV moves, the background varies, which hinders accurate detection [1].

Traditionally, motion-based methods (e.g., optical flow) have been used for background subtraction, from which vehicles on a highway might be detected and counted [11]–[13]. However, as these methods require the vehicles to be in motion, applications are limited, and implementing these methods on moving frameworks, such as UAVs, can be challenging [1].

The majority of previous research has focused on extracting hand-crafted features, such as the scale-invariant

feature transform (SIFT), speeded-up robust features (SURF), the histogram of oriented gradients (HOG), and Haar-like features, which are then followed by a variety of classifiers, such as support vector machines or Adaptive Boosting (AdaBoost) to detect vehicles [1], [14]–[19]. These methods are based on sliding window and multi-scale searches, which are computationally expensive (the former especially so) [20].

In recent studies, deep learning-based methods outperformed previous approaches, particularly for computer vision and scene understanding tasks [20]–[22]. 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 [3]. CNN-based object detectors are mainly divided into two-step and one-step detectors. Two-step detectors, such as R-CNNs [23], Fast R-CNN [24], Faster R-CNN [25] and Mask R-CNN [26], use region proposals to complete object location regression and classification processes in two steps. In contrast, one-step detectors, such as YOLOv3 [27] and the single shot multibox detector (SSD) [28], predict object locations and classes simultaneously in a single network. Hence, one-step detectors provide faster detection than two-step detectors [9].

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 [21]. In addition, training these networks generally demands a high computational cost, and the lack of well-annotated training data adds to the challenge [10], [22].

In this study, we aim to introduce a robust vehicle detection model that requires limited training data and computational power. Employing a modified YOLOv3 as the detector network, we combined various state-of-the-art pre-trained classification networks to compare performance. Additionally, deep neural network (DNN) performance was improved with a consecutive fully connected neural network (fcNN) that used 3D depth features; depth maps were generated using a pair of aerial images at each instance [29]. Aerial images were obtained using a UAV over two harbor areas on different days. The collected data were then annotated and used for training and evaluation of the proposed method. The main contributions of this article are as follows.

1) A modified YOLOv3 detector network was employed using hierarchical features at two different layers to utilize spatial information to detect numerous vehicles on the ground.
2) Several pre-trained feature extractor networks were implemented to evaluate their performance according to their number of layers, size, and processing time.
3) 3D depth maps were utilized by an fcNN to increase the precision of the DNN output. The experimental results indicate that three-dimensional features efficiently improved performance.

The manuscript is organized as follows. Following Section I, in Section II the related works are reviewed. Afterward, in Section III the methodology is described. Then, the experimental results and discussion are provided in Section IV. Finally, Section V discusses the key findings of this work.

II. RELATED WORKS

A broad range of methods have been proposed in the literature for vehicle detection in aerial images. In this section, we briefly introduce some of the CNN-based methods.

As mentioned earlier, CNNs are first employed for feature extraction. Various detector networks, such as Fast R-CNN, Faster R-CNN, and SSD, are then used to detect vehicles in the scene. Tang *et al.* [30] introduced a method based on Faster R-CNN that uses a hyper region proposal network with a combination of hierarchical features to improve small vehicle detection. Similarly, the accurate-vehicle-proposal-network [21] uses the same concept of hierarchical feature maps integration. In [7], a cascade of two independent CNNs was employed to utilize shallow and deep feature maps. Moreover, [1] presents a one-step detector SSD method that implements a ResNet network [31] for feature extraction. Furthermore, [9] reports a feature fusion and scaling-based SSD method in which a deconvolution module and an average pooling layer successively improved feature map resolution. Inspired by these methods, we used a one-step detector YOLOv3 that integrates hierarchical features from two layers based on the prior knowledge of the dimensions of heavy vehicles in the scene.

Meanwhile, other methods have utilized segmentation to prevent false positives and improve performance. In [32], image segmentation was conducted to detect homogeneous regions. Next, a CNN was used to extract features from these regions and then a linear support vector machine was used for classification. In a similar approach, [33] utilized a fully convolutional network for segmentation and vehicle detection, followed by a CNN for vehicle classification. Finally, in [10], superpixel segmentation was implemented to obtain spatially encoded features that were then combined with deep hierarchical features to improve detection performance. In the proposed method, we considered a similar strategy of false positives reduction, however, we utilized 3D features instead of segmentation.

Tayara *et al.* [3] proposed a regression model that used a fully convolutional regression network to construct a density map; applying an empirical threshold to the output returned the count and location of the detected vehicles. In [34], a unified residual fully convolutional network was presented for vehicle detection. By combining feature representations from various residual blocks, the network was able to predict the semantic segmentation masks and semantic boundaries of the vehicle regions simultaneously. However, these methods required high-resolution aerial images and were time-consuming during inference.

[22] investigated the effect of training data on vehicle detection. By applying hard example mining to stochastic gradient descent (SGD), examples with the largest losses

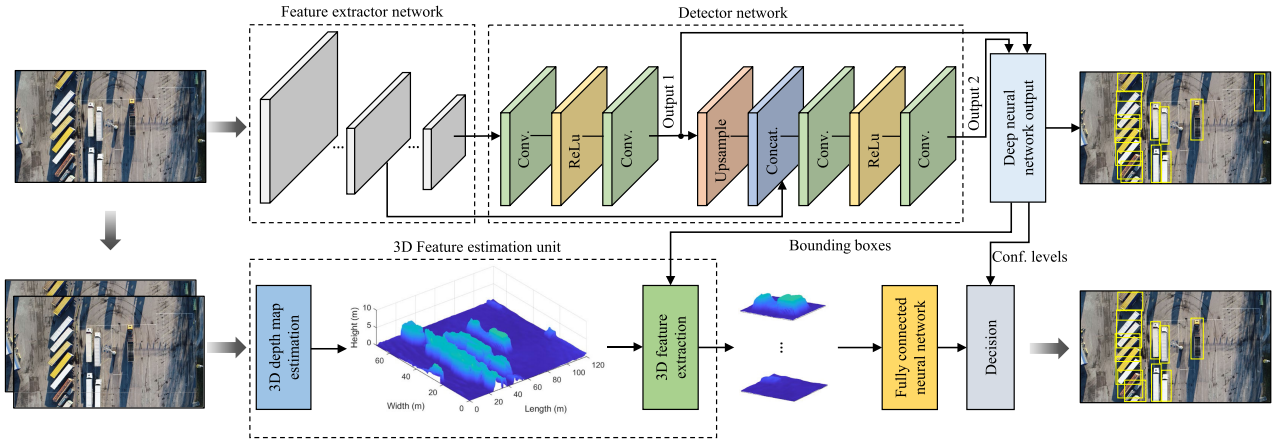


FIGURE 1. Overview of the proposed system based on a DNN trained on aerial images and an fcNN trained on depth maps for vehicle detection.

were utilized as the training data. In addition, [35] suggested using scale-adaptive anchor boxes to match the size distribution of the training data. Furthermore, they proposed a circular flow inspired by the attention mechanism for feature extraction. Similarly, the anchor boxes in our proposed network were set by clustering the bounding boxes of the annotated training data.

Vehicle orientation information is also particularly important for vehicle detection in dense scenes and in other applications, such as tracking, that might require the vehicle's trajectory. Li *et al.* [20] proposed a trainable network called a rotatable residual network (R^3-Net) to generate rotatable bounding boxes using a rotatable region proposal network; a rotatable detection network was then used for classification and regression of the regions of interest. However, annotating the training data with orientation information is computationally more expensive.

According to the related works, one-step detectors can be considered to provide a fast performance [1], [9]. Furthermore, the size of the targets should be taken into account in the integration of hierarchical features process in order to preserve spatial information [21], [30]. The anchor boxes also need to be selected based on the knowledge of the size distribution of the targets [22]. In the method presented in this article, these concepts were considered and implemented. However, the main idea in this work that was not investigated in the previous approaches, is to utilize information complementary to the image features such as 3D features. To do so, we propose a trainable structure comprising a DNN to detect and localize vehicles in aerial images and an fcNN based on three-dimensional depth maps to refine performance.

III. METHODOLOGY

In this section, the methodology of the proposed system based on deep learning and 3D depth maps is described (see Figure 1). This system is intended to detect heavy vehicles such as trucks and semi-trailers in aerial images, mostly in harbor areas. This application was considered to better understand the transportation parameters in the local industrial harbors for future planning. However, this approach can be implemented for other object detection applications by utilizing 3D features as complementary to images.

First, for the feature extractor network we modified and implemented several CNN architectures as the base network, including DarkNet-53 [27], SqueezeNet [36], MobileNet-v2 [37] and DenseNet-201 [38]. From these networks, hierarchical feature maps from various layers were integrated to preserve spatial information and improve detection performance. Efficiency, processing time, and memory required were the factors by which these feature extractor networks were evaluated. DarkNet-53, MobileNet-v2, and SqueezeNet are among the more recent CNNs; with relatively few parameters, they require less processing time and memory. On the other hand, DenseNet-201 has better feature propagation, but at the cost of more parameters and a larger size. The acquired feature maps were then applied to a YOLOv3 detector network to provide predictions across different scales. YOLOv3 was selected as the detector network due to its high speed and the flexibility that can be associated with this network for accurate detection [9].

Simultaneously, a depth map unit was employed to produce 3D features of the area using pairs of aerial images [29]. Conventional DNN analysis of aerial images does not utilize height information from 3D features and objects; however, these features can contain distinctive information, particularly for object detection in aerial images. Subsequently, the 3D feature maps of the DNN-detected regions were analyzed by an fcNN to improve the precision of the final detection.

A. FEATURE EXTRACTOR NETWORKS

The base networks for feature extraction were all pre-trained on more than a million images from ImageNet [39], as this accelerates training and provides powerful feature representation for the detector network. Although YOLOv3 was originally proposed based on DarkNet-53 [27], in this study we modified a variety of networks to extract features that we then applied to YOLOv3. This was done to take advantage of the distinct characteristics of each base network, such as memory, number of operations, accuracy, and speed, and select

TABLE 1. Feature Extractor Networks Architectures Utilized in This Work for Feature Representation of Aerial Images.

DarkNet-53												
Layers	conv	conv	conv conv res	conv	conv conv res	conv	conv conv res	conv	conv conv res	conv	conv conv res	classification layer
Output size	256×256	128×128	128×128	64×64	64×64	32×32	32×32	16×16	16×16	8×8	8×8	1×1
Filters	3×3, 32, st 1	3×3, 64, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times1$	3×3, 128, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times2$	3×3, 256, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times8$	3×3, 512, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times8$	3×3, 1024, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times4$	avg pool, 1000d fc, softmax
SqueezeNet												
Layers	conv1	maxpool1	fire2 fire3	maxpool3	fire4 fire5	maxpool5	fire6 fire7	fire8 fire9	conv10	classification layer		
Output size	113×113	56×56	56×56	28×28	28×28	14×14	14×14	14×14	14×14	1×1		
Filters	3×3, 64, st 2	3×3, 64, st 2	$\begin{bmatrix}1\times1\\1\times1\\3\times3\end{bmatrix}\times2$	3×3, 64, st 2	$\begin{bmatrix}1\times1\\1\times1\\3\times3\end{bmatrix}\times2$	3×3, 256, st 2	$\begin{bmatrix}1\times1\\1\times1\\3\times3\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\\1\times1\\3\times3\end{bmatrix}\times2$	1×1, 1000, st 1	avg pool, softmax		
MobileNet-v2												
Layers	conv2d	bottleneck1	bottleneck2	bottleneck3	bottleneck4	bottleneck5	conv2d	classification layer				
Output size	112×112	112×112	56×56	28×28	14×14	7×7	7×7	1×1				
Filters	3×3, 32, st 2	$\begin{bmatrix}3\times3\\1\times1\end{bmatrix}\times1$	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}\times1$	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}\times7$	$\begin{bmatrix}1\times1\\3\times3\\1\times1\end{bmatrix}\times6$	7×7, 1280, st 1	avg pool, 1000d fc, softmax				
DenseNet-201												
Layers	conv	maxpool	dense1	transition1	dense2	transition2	dense3	transition3	dense4	classification layer		
Output size	112×112	56×56	56×56	56×56 28×28	28×28	28×28 14×14	14×14	14×14 7×7	7×7	1×1		
Filters	7×7, 64, st 2	3×3, 64, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times6$	1×1, 128, st 1 2×2 avg pool, 128, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times12$	1×1, 256, st 1 2×2 avg pool, 256, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times48$	1×1, 896, st 1 2×2 avg pool, 896, st 2	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times32$	avg pool, 1000d fc, softmax		

the specific characteristics best suited for this application. Feature maps were integrated from two layers according to truck size in the scene, as the target object in the scene, and the remaining layers were removed. Later layers in the extractor network represent more complicated features while losing spatial information [40]. Therefore, the prior knowledge of the target's size in the image is important to extract meaningful features as well as preserving spatial information. The final layers in the extractor network were originally employed for classification tasks, that in this work, they were not utilized and hence they were removed.

As mentioned earlier, DarkNet-53 was originally introduced as the base network of YOLOv3. In this study, we applied this network as one of the feature extractor networks. DarkNet-53 is a combination of DarkNet-19 and the residual network connections, which is significantly larger than its predecessor (see Table 1). The input size of the network is 256×256; the layers for feature extraction were "res," with an output size of $32\times32\times256$ and "res," with an output size of $16 \times 16 \times 512$. The reduced network contains 149 layers and 167 connections.

Another network studied in this work was SqueezeNet [36], which can achieve AlexNet [41] accuracy with far fewer parameters. A smaller number of parameters can be crucial for hardware deployment and real-time performance. SqueezeNet configuration is presented in Table 1. The minimum required network input size is 227×227; the selected feature layers of the SqueezeNet network were "fire5," with an output size of $28 \times 28 \times 256$ and "fire9," with an output size of $14\times14\times512$. The reduced network contains 62 layers and 69 connections.

MobileNet-v2 [37] is another memory-efficient network. This network is based on inverted residual structure in which thin bottleneck layers are connected. At each module, a low-dimensional compressed feature representation is first expanded, then filtered by a lightweight depthwise convolution, and finally projected back to a low-dimensional representation (see Table 1). The network input size is 224×224. The hierarchical feature maps were extracted from the "bottleneck3" layer, with an output size of $28 \times 28 \times 192$ and the "bottleneck4" layer, with an output size of $14 \times 14 \times 576$. The reduced network contains 116 layers and 123 connections.

Finally, DenseNet-201 [38] was the last feature extractor network implemented in this study. Each layer in this network is connected to the following layers. Hence, the input of each layer contains the feature maps of all the preceding layers (see Table 1). This improves feature propagation and reduces the fading gradient problem in deep networks, which leads to more efficient training [38]. The network input size is 224×224. The feature maps were extracted from "dense2" layer, with an output size of $28 \times 28 \times 512$ and "dense3" layer, with an output size of $14 \times 14 \times 1152$. The reduced network contains 475 layers and 540 connections.

B. DETECTOR NETWORK

For this study, we employed a detector network based on a YOLOv3 model and trained it for vehicle detection [27]. First, anchor boxes were set as priors for the network on which to perform predictions. Anchor boxes were selected using k-means clustering based on the bounding boxes of the annotated training data. The distance metric between bounding boxes for clustering is,

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid}), \tag{1}$$

where IOU is defined as the intersection over union between each bounding box and a cluster centroid using their width and height features [42]. Although higher numbers of clusters could be expected to lead to higher recall, model complexity (e.g. network size, required convolutions, training time) would also be increased. Considering these factors, 6 was selected as the number of clusters, which is three anchor boxes per scale.

For each bounding box, the network predicts t_x, t_y as the offsets from the center of each grid cell and t_w, t_h as the offsets from the width and height of the anchor box [27]. These predictions correspond to the position and size of the bounding box relative to the entire image as,

$$b_x = \sigma(t_x) + c_x \tag{2}$$
$$b_y = \sigma(t_y) + c_y \tag{3}$$
$$b_w = p_w e^{t_w} \tag{4}$$
$$b_h = p_h e^{t_h} \tag{5}$$

where c_x and c_y are grid cell offset from the top-left corner of the image, $\sigma()$ is a sigmoid function and p_w and p_h are the relative width and height of the anchor box (the bounding box prior). The ground truth offset values $\hat{t}_x$, $\hat{t}_y$, $\hat{t}_w$, and $\hat{t}_h$ can be computed using the above equations.

The loss function used for optimization comprises the coordinate loss, objectness loss, and class predication loss [27]. First, the coordinate loss during training is computed as the sum of squared error as,

$$L_{\text{coord}} = \lambda_{\text{coord}}\Big(\sum_{i=0}^{N^2}\sum_{j=0}^{B} k_{ij}^{\text{obj}}[(t_x - \hat{t}_x)^2 + (t_y - \hat{t}_y)^2] + \sum_{i=0}^{N^2}\sum_{j=0}^{B} k_{ij}^{\text{obj}}[(t_w - \hat{t}_w)^2 + (t_h - \hat{t}_h)^2]\Big), \tag{6}$$

where N^2 is the number of grid cells in the image, B is the number of bounding boxes per cell, λ_{coord} is the penalizing weight for coordinate loss, and k_{ij}^{obj} denotes if the j^{th} bounding box in the i^{th} cell contains an object.

In addition, the network computes an objectness value for each bounding box using logistic regression. In each cell, if an anchor box overlaps an object more than a threshold and more completely than other anchor boxes, then this value should be 1. In this way, only one anchor box is assigned to an object. The ground truth value for objectness is 1 if there is an object in the cell, and 0 if not. Thus, objectness loss is computed as,

$$L_{\text{obj}} = \sum_{i=0}^{N^2}\sum_{j=0}^{B} k_{ij}^{\text{obj}}[-\log(\sigma(t_o))] + \lambda_{\text{noobj}} \sum_{i=0}^{N^2}\sum_{j=0}^{B}(1 - k_{ij}^{\text{obj}})[-\log(1 - \sigma(t_o))], \tag{7}$$

where λ_{noobj} is penalizing weight for false positives and $\sigma(t_o)$ is the predicted objectness.

Finally, the network predicts the class label of each bounding box using independent logistic classifiers. Class loss is obtained based on binary cross-entropy loss,

$$L_{\text{class}} = \sum_{i=0}^{N^2}\sum_{j=0}^{B} k_{ij}^{\text{obj}}\sum_{c=1}^{C}[\text{BCE}(\hat{y}_c, \sigma(s_c))], \tag{8}$$

where BCE represents binary cross-entropy, $\hat{y}_c$ is the ground truth label, $\sigma(s_c)$ is the prediction class, and C is the total number of classes. Thus, the total loss is computed as the summation of the above three losses,

$$L = L_{\text{coord}} + L_{\text{obj}} + L_{\text{class}}. \tag{9}$$

The original YOLOv3 network takes feature maps from three layers to incorporate finer-grained information from the earlier features and more meaningful semantic information from the later ones [27]. However, in this work, feature maps were integrated from two layers to take advantage of fine-grained as well as meaningful features without the final layer. It is due to the relatively small size of the vehicles in the aerial image that requires more preservation of spatial information. Furthermore, the original YOLOv3 architecture employs nine anchor boxes and performs prediction on 80 classes, while the proposed network was based on six anchor boxes (three per scale) with one class prediction. The remaining parameters such as the overlap threshold and penalizing weights were kept the same as proposed in the original network.

In this work, the detector network employs He initialization [43] method to initialize its weights. Then, the pre-trained extractor network along with the initialized detector network were trained using the proposed training data. Here, an SGD optimizer was used to update network weights.

C. 3D DEPTH MAP ESTIMATION

In this work, 3D depth maps were utilized to improve deep network vehicle detection. As 3D features may contain crucial information to distinguish vehicles from other objects in the scene, height features from the depth maps were fused with image features represented by the DNN for improved performance.

In addition, the depth map in aerial images is particularly important in transportation applications, as all the objects have a height above the reference surface (such as the road). Hence, in aerial images with a top-down view, the distance of

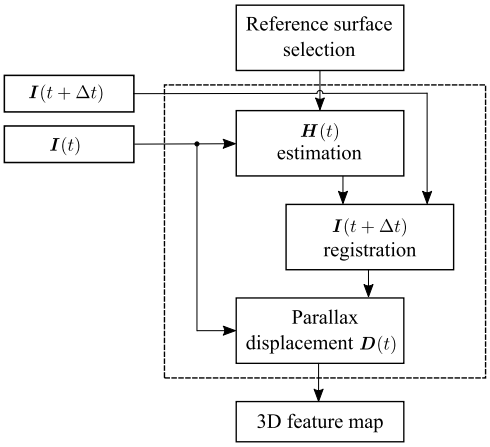


FIGURE 2. Estimation of a 3D depth map of a scene using a pair of aerial images ($I(t)$, $I(t+\Delta t)$) where $H(t)$ and $D(t)$ are the transformation matrix and the parallax displacement, respectively [29].

objects from the camera can be correlated with their classes, which does not necessarily apply in natural scenes imagery.

To add 3D information to the detection process, the 3D depth map was first generated using a pair of aerial images. Upon registration of the aerial images with slightly different viewpoints based on a reference surface, the parallax displacement of a pixel indicates the height of that point in the real-world coordinate system [29] (see Figure 2).

In this process, orthogonal pairs of aerial images were obtained by a moving UAV where the baseline between two views was parallel to the direction of the movement. The sampling interval, T (s) was computed based on the constant speed of the UAV, v_{UAV} (m/s), the flight altitude, h_{UAV} (m), the ground sampling distance, GSD (m/px), and the minimum detection height, h_{min} (m). Consequently, after alignment of the aerial images with respect to the ground as the reference surface, the corresponding height of each pixel was calculated as,

$$h = \frac{GSD \cdot h_{\mathrm{UAV}} \cdot d}{GSD \cdot d + T \cdot v_{\mathrm{UAV}}} \tag{10}$$

where d (px) is the parallax displacement of that pixel [29].

D. FULLY CONNECTED NEURAL NETWORK

After generating the 3D depth maps, an fcNN was trained based on the 3D features of heavy vehicles such as trucks and semi-trailers. The proposed system initially detects targets in the aerial images using the trained DNN while a 3D feature map of the scene is generated using a pair of aerial images. Next, the coordinates of the detected bounding boxes obtained by the DNN are used to extract the 3D features of the candidates from the scene. These 3D features which correspond to detection results in DNN, are then fed to the trained fcNN to detect heavy vehicles.

The fcNN was initially trained using 3D feature representations of positive and negative samples. The architecture of the fcNN was designed to have one hidden layer with 20 units. The number of input features was set to 50 and hence, all samples were first downsampled. In addition, the proposed

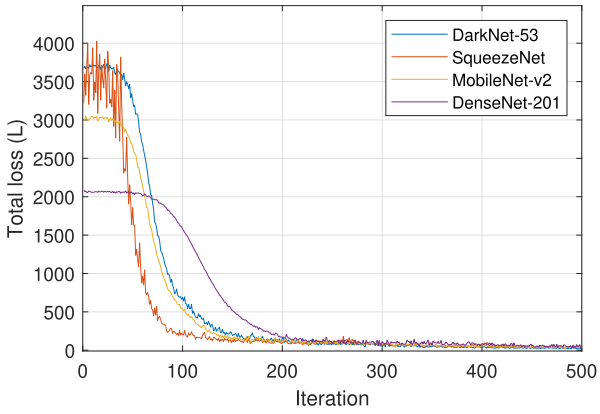


FIGURE 3. Total loss of DNNs during training in the first 500 out of 3000 iterations with various base networks using aerial images.

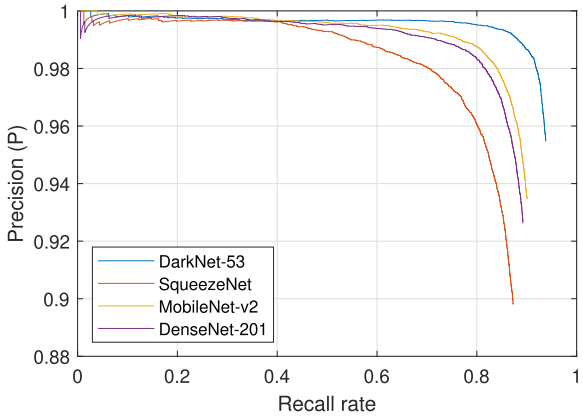


FIGURE 4. The precision-recall curves of DNNs with different feature extractors on the test set of aerial images.

fcNN employed sigmoid as activation function for the outputs of hidden units and mean squared error as the loss function. Finally, the network was optimized by using the Levenberg-Marquardt algorithm [44].

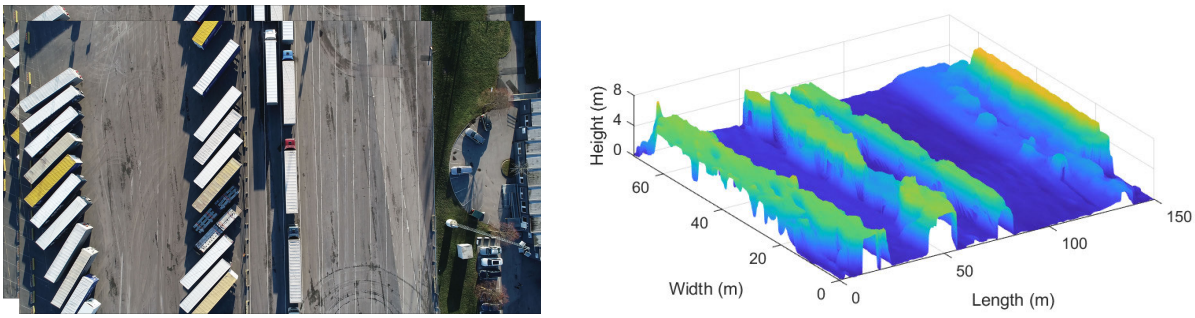
E. DECISION CRITERIA

The final decision criteria were based on the confidence levels of targets detected by the DNN using aerial images and the successive fcNN using 3D features. The DNN was initially applied to detect the targets in the aerial image and then, the fcNN was implemented to evaluate detection results from the DNN. If the DNN confidence level is between 0.05 and 0.5 (or between 0.01 and 0.5), then the confidence level of the fcNN output based on 3D features of that region is considered to evaluate the target. Otherwise, if the DNN confidence level of the region is greater than 0.5, it is considered as the target (without the fcNN intervention). In this way, majority of false positives from the deep learning detection results can be identified; therefore, precision is improved by utilizing the corresponding 3D features.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. DATA SETS AND EVALUATION PARAMETERS

The data sets for training and testing of the proposed system were obtained using a UAV over two industrial harbors from



(a) (b)

FIGURE 5. The 3D depth map of a scene based on parallax displacement: (a) a pair of aerial images and (b) the computed 3D depth map.

17 flights in over 14 days with various lighting conditions [45]. The aerial images were orthogonal and covered the parking space of the harbors where passenger vehicles and trucks waited to embark onto ships. These sites are often densely occupied with parked trailers, and counting and tracking their numbers is important. The images comprised 3840 × 2160 pixels and were annotated according to trucks and trailers to construct the ground truth. Data sets from 14 flights were used to develop the DNN to detect the target regions; data from three flights were used to build the fcNN based on the 3D depth maps. Finally, the proposed system was tested on previously unseen aerial images that had been obtained from three other flights over three days.

The criteria used to evaluate the performance of the proposed vehicle detection system were recall rate R, precision rate P, and F1-score $F1$,

$$R = \frac{N_{tp}}{N_{tp} + N_{fn}}, \quad P = \frac{N_{tp}}{N_{tp} + N_{fp}}, F1 = 2 \cdot \frac{R \cdot P}{R + P} \tag{11}$$

where N_{tp}, N_{fn}, and N_{fp} are the number of true positives, false negatives, and false positives, respectively.

B. TRAINING DEEP NEURAL NETWORKS

As described earlier, the DNN consists of a feature extractor network followed by a detector network, YOLOv3. Four networks (DarkNet-53, SqueezeNet, MobileNet-v2, and DenseNet-201) were utilized as the base network for feature representation. A total of 3023 aerial images were used, including 1813 images for training and 1210 images for testing the deep networks. The images were downsampled from 3840 × 2160 pixels to the dimensions required by the respective networks. The SGD method was adopted to optimize the networks. The learning rate was initially set to 0.001. This was reached exponentially after 1000 iterations, and then the rate was dropped in two steps, after 2200 and 2800 iterations, to 0.0001 and 0.00001, respectively. Each network was trained over 3000 iterations with a mini-batch size of 8, and the L2 regularization factor was set to 0.0005.

The objective of training is to minimize the total loss, which consists of coordinate loss, objectness loss, and class loss. The total loss during training of the networks with various feature extractors are presented in Figure 3. To better demonstrate the differences, only the loss in the first 500 iterations is illustrated in the graph. Network performance was evaluated by running trained detectors on each image in the test set, from which the precision-recall curves were obtained (see Figure 4).

TABLE 2. The Results of Training and Testing DNNs With Various Base Networks for Detection of Heavy Vehicles in Aerial Images.

Networks	Mean IoU	Training time (sec)	Memory size (MB)	Average precision
DarkNet-53 + YOLOv3	0.81	4785	69	93.4 %
SqueezeNet-53 + YOLOv3	0.72	2817	9	86.1 %
MobileNet-v2 + YOLOv3	0.72	4763	18	89.5 %
DenseNet-201 + YOLOv3	0.75	8710	179	88.5 %

The results show that the DNN containing DarkNet-53 achieved the highest average precision (AP); it was therefore selected as the extractor network for the proposed system. However, other networks also performed well, particularly taking into consideration the limited training time and memory available (see Table 2).

C. 3D DEPTH MAP GENERATION

As explained earlier, the 3D depth maps of scenes are generated using a pair of orthogonal aerial images. After registration, the parallax displacement of each pixel demonstrates the height of that point in the real-world coordinate system. In this work, the aerial images were obtained using a UAV traveling with a speed of $v_{\text{UAV}} = 5.07$ m/s and height of $h_{\text{UAV}} = 100$ m. In addition, the minimum detection height, the camera's ground sampling distance and the sampling interval to obtain a pair of aerial images were set at $h_{\text{min}} = 0.33$ m, $GSD = 0.039$ m/px and $T = 2.3$ s, respectively. Accordingly, the height of each pixel was calculated based on the computed parallax displacement and the aforementioned parameters [29] (see Figure 5).

D. TRAINING A FULLY CONNECTED NEURAL NETWORK

A fully connected network was trained using the computed 3D depth maps to improve the performance of deep networks. The data set used to train and test the neural network was acquired from nine depth maps containing 187 positive and negative samples (see Figure 6). As described earlier, the neural network had one hidden layer comprising 20 units, and the input data were downsampled to 50 features per sample.

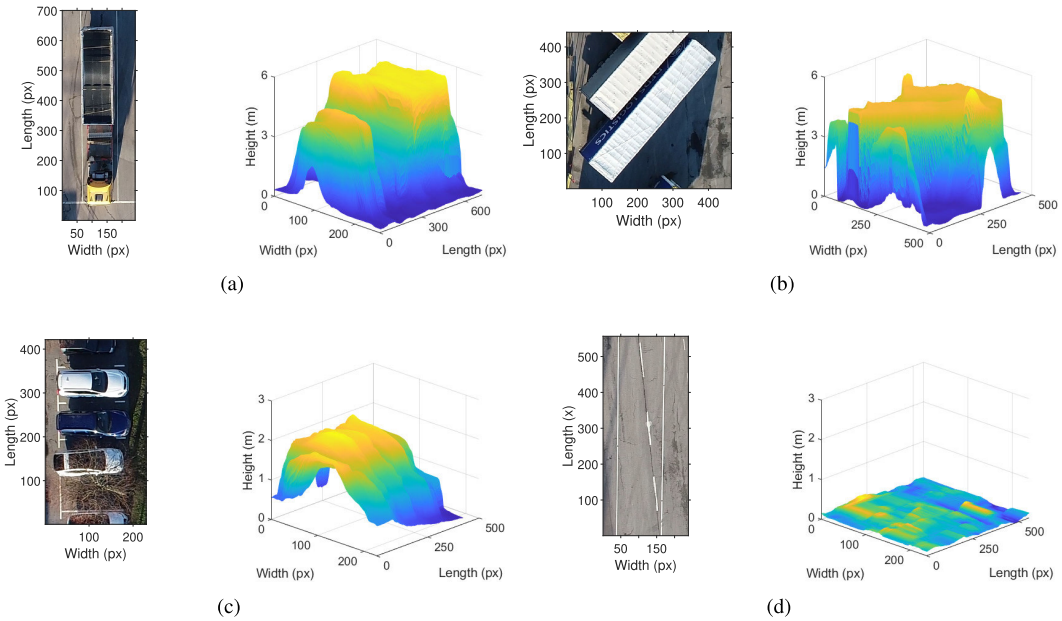


FIGURE 6. Examples of the 3D features used for training the fcNN and their corresponding regions in the aerial image: (a) and (b) represent positive samples such as trucks and trailers; (c) and (d) represent negative samples.

TABLE 3. Performance Evaluation of the Proposed System With Various DNN Confidence Thresholds (CT) and Utilizing 3D Feature Maps.

Evaluation parameters	DNN CT = 0.01		DNN CT = 0.05	
	DNN	DNN+fcNN	DNN	DNN+fcNN
True positive (N_{tp})	135	135	134	134
False negative (N_{fn})	11	11	12	12
False positive (N_{fp})	18	5	4	0
Recall (R)	92.46 %	92.46 %	91.78 %	91.78 %
Precision (P)	88.23 %	96.43 %	97.10 %	100 %
F1-score	90.30 %	94.40 %	94.36 %	95.72 %

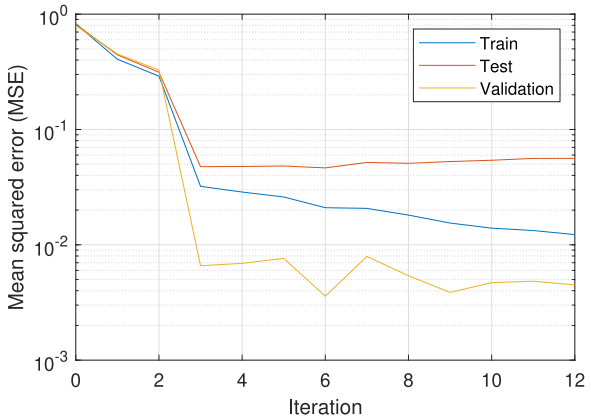


FIGURE 7. Performance plot of the fcNN as based on the 3D depth features. The best validation performance was achieved on the sixth iteration.

To optimize the network, the weights and bias values were updated according to the Levenberg-Marquardt algorithm [44] (see Figure 7).

E. PERFORMANCE EVALUATION

Finally, the proposed system, including the trained DNN based on DarkNet-53 and YOLOv3 and the trained fcNN based on 3D features, was evaluated. To do so, aerial images were acquired from videos of three UAV flights over three days. These images were previously unseen and had not been employed in any earlier steps. Subsequently, nine orthogonal aerial images were used as input for the DNN. A total of 146 targets, consisting of trucks, semi-trailers, and trailers, were presented in these aerial images. In addition, for each scene, a 3D depth map was generated using parallax displacement from a pair of aerial images.

The default confidence threshold for the DNN was 0.5; detected bounding boxes with higher confidence levels were considered as targets. Consequently, decreasing the confidence threshold would cause the deep network to detect more targets and more false positives. In other words, while the recall rate could be improved by lowering the threshold, detection precision would be hindered. To correctly identify targets within the bounding boxes detected by the DNN with confidence scores below 0.5, 3D feature maps of those areas were extracted and fed to the trained fcNN. Performance of the proposed system with and without utilizing the 3D features is presented in Table 3.

According to the results, the combination of the DNN trained on aerial images and the fcNN trained on 3D feature maps had the best performance in terms of F1-score. The precision of the DNN with the confidence threshold of 0.05 was improved by 2.9 % by utilizing the 3D feature maps; increasing the F1-score to 95.72 %. Similarly, the precision and

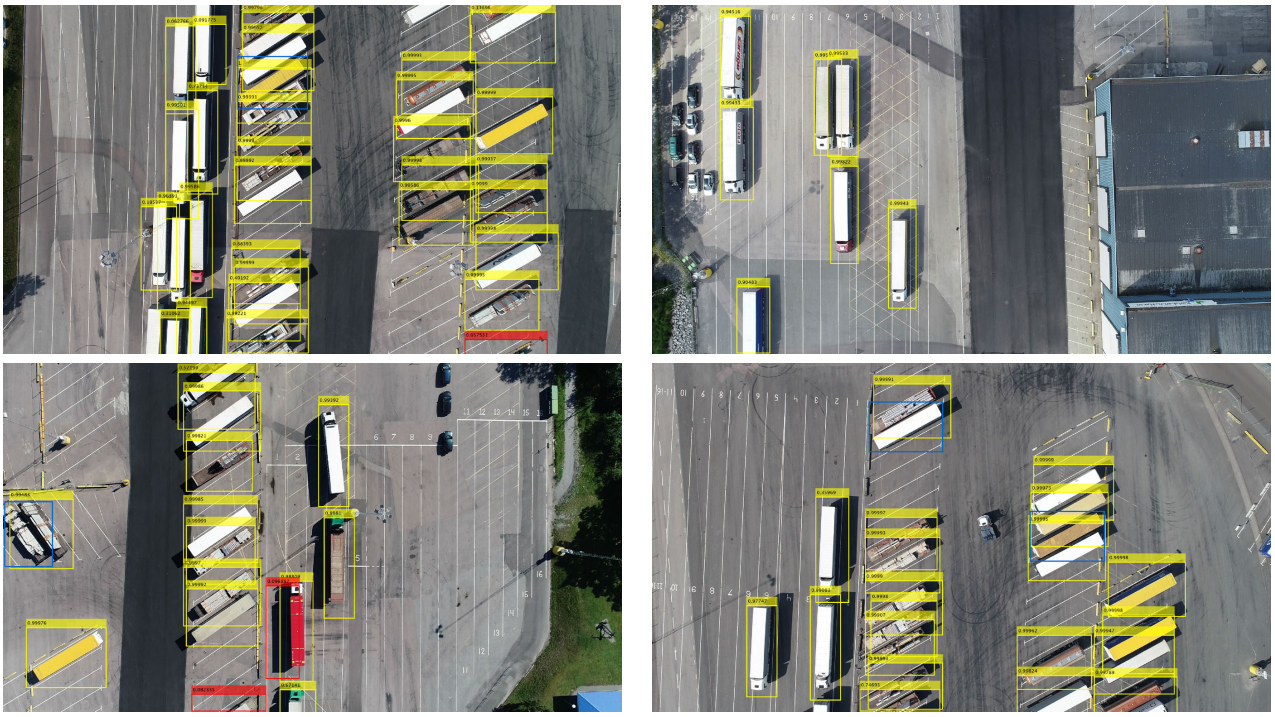


FIGURE 8. Examples of the detection performance of the proposed system. Yellow bounding boxes are the final results, red bounding boxes indicate false positives that were removed by the fcNN using 3D features, and blue bounding boxes reflect missed targets.

F1-score of the DNN with a 0.01 confidence threshold were improved by 8.2 % and 4.1 %, respectively. These results demonstrate that introducing 3D features to refine the detection results obtained from the DNNs can noticeably improve detection precision. Therefore, fusing images and 3D features is crucial to achieve more reliable detection results (see Figure 8).

V. CONCLUSION

In this article, a novel approach to vehicle detection in aerial images based on deep neural networks and 3D feature maps is presented. A modified YOLOv3 detector network with various base networks, including DarkNet-53, SqueezeNet, MobileNet-v2 and DenseNet-201, were employed to detect trucks, semi-trailers, and trailers. The properties and characteristics of these network architectures were studied. This study was experimentally conducted in real-world conditions and in the practical application of parking spaces of industrial harbors where passenger vehicles and trucks lined up to embark onto ships. The results show that although DarkNet-53 achieved the highest average precision of 93.4 %, other networks also performed satisfactorily, particularly when considering the constraints of processing time and memory. Next, the role of 3D features was studied to improve DNN performance. An fcNN was trained using 3D features and placed in cascade with the DNN. The experimental results demonstrate that utilizing 3D depth maps improved the precision of the DNN substantially, obtaining an F1-score of 95.72 %. It can be concluded that 3D features improve the performance of vision-based deep neural networks. Future research should develop a unified deep neural network that includes 3D features as part of the input signal.

ACKNOWLEDGMENT

The authors would like thank the Port of Karlshamn, the Port of Karlskrona, NetPort Science Park, the Municipality of Karlshamn, the Swedish Transport Agency, and the Swedish Transport Administration for their support in this work. They would also like to thank M. Rameez from BTH for the valuable discussions.

REFERENCES

[1] 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, doi: 10.1109/JSTARS.2018.2879368.

[2] 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, doi: 10.1109/LGRS.2015.2439517.

[3] H. Tayara, K. G. Soo, and K. T. Chong, "Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network," *IEEE Access*, vol. 6, pp. 2220–2230, 2018, doi: 10.1109/ACCESS.2017.2782260.

[4] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 339–351, Feb. 2015, doi: 10.1109/TCSVT.2014.2339592.

[5] M. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, "A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery," *Remote Sens.*, vol. 9, no. 2, p. 100, Jan. 2017, doi: 10.3390/rs9020100.

[6] M. Pettersson, M. Dahl, V. T. Vu, and S. Javadi, "Future satellite and drone monitoring of the baltic-adriatic corridor, harbors, and motorways of the sea," Swedish Digit. Sci. Arch. (DiVA), Blekinge Inst. Technol., Karlskrona, Sweden, Tech. Rep., 2019, doi: 10.13140/RG.2.2.23511.21920.

[7] 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, doi: 10.3390/s17122720.

[8] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 92, pp. 79–97, Jun. 2014, doi: 10.1016/j.isprsjprs.2014.02.013.

[9] X. Liang, J. Zhang, L. Zhuo, Y. Li, and Q. Tian, "Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1758–1770, Jun. 2020, doi: 10.1109/TCSVT.2019.2905881.

[10] H. Long, Y. Chung, Z. Liu, and S. Bu, "Object detection in aerial images using feature fusion deep networks," *IEEE Access*, vol. 7, pp. 30980–30990, 2019, doi: 10.1109/ACCESS.2019.2903422.

[11] L. Wang, F. Chen, and H. Yin, "Detecting and tracking vehicles in traffic by unmanned aerial vehicles," *Autom. Construction*, vol. 72, pp. 294–308, Dec. 2016, doi: 10.1016/j.autcon.2016.05.008.

[12] G. Mo and S. Zhang, "Vehicles detection in traffic flow," in *Proc. 6th Int. Conf. Natural Comput.*, Yantai, China, Aug. 2010, pp. 751–754, doi: 10.1109/ICNC.2010.5583178.

[13] M. Dahl and S. Javadi, "Analytical modeling for a video-based vehicle speed measurement framework," *Sensors*, vol. 20, no. 1, p. 160, Dec. 2019, doi: 10.3390/s20010160.

[14] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1635–1647, Mar. 2014, doi: 10.1109/TGRS.2013.2253108.

[15] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images," *Sensors*, vol. 16, no. 8, p. 1325, Aug. 2016, doi: 10.3390/s16081325.

[16] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, Jun. 2011, doi: 10.1109/TPAMI.2010.182.

[17] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014, doi: 10.1109/LGRS.2014.2309695.

[18] H.-Y. Cheng, C.-C. Weng, and Y.-Y. Chen, "Vehicle detection in aerial surveillance using dynamic Bayesian networks," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2152–2159, Apr. 2012, doi: 10.1109/TIP.2011.2172798.

[19] Z. Chen, C. Wang, C. Wen, X. Teng, Y. Chen, H. Guan, H. Luo, L. Cao, and J. Li, "Vehicle detection in high-resolution aerial images via sparse representation and superpixels," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 103–116, Jan. 2016, doi: 10.1109/TGRS.2015.2451002.

[20] Q. Li, L. Mou, Q. Xu, Y. Zhang, and X. X. Zhu, "R3-Net: A deep network for multioriented vehicle detection in aerial images and videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5028–5042, Jul. 2019, doi: 10.1109/TGRS.2019.2895362.

[21] Z. Deng, H. Sun, S. Zhou, J. Zhao, and H. Zou, "Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3652–3664, Aug. 2017, doi: 10.1109/JSTARS.2017.2694890.

[22] Y. Koga, H. Miyazaki, and R. Shibasaki, "A CNN-based method of vehicle detection from aerial images using hard example mining," *Remote Sens.*, vol. 10, no. 1, p. 124, Jan. 2018, doi: 10.3390/rs10010124.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[24] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[25] 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, doi: 10.1109/TPAMI.2016.2577031.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[27] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[28] W. Liu *et al.*, "SSD: Single shot multiBox detector," 2015, *arXiv:1512.02325*. [Online]. Available: https://arxiv.org/abs/1512.02325

[29] S. Javadi, M. Dahl, and M. I. Pettersson, "Change detection in aerial images using three-dimensional feature maps," *Remote Sens.*, vol. 12, no. 9, p. 1404, Apr. 2020, doi: 10.3390/rs12091404.

[30] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors*, vol. 17, no. 2, p. 336, Feb. 2017, doi: 10.3390/s17020336.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[32] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, Mar. 2017, doi: 10.3390/rs9040312.

[33] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sens.*, vol. 9, no. 4, p. 368, Apr. 2017, doi: 10.3390/rs9040368.

[34] L. Mou and X. X. Zhu, "Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6699–6711, Nov. 2018, doi: 10.1109/TGRS.2018.2841808.

[35] 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, doi: 10.1109/TIE.2019.2899548.

[36] F. Iandola *et al.*, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360v4*. [Online]. Available: https://arxiv.org/abs/1602.07360v4

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013, *arXiv:1311.2901*. [Online]. Available: https://arxiv.org/abs/1311.2901

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[42] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*. [Online]. Available: https://arxiv.org/abs/1612.08242

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," 2015, *arXiv:1502.01852*. [Online]. Available: https://arxiv.org/abs/1502.01852

[44] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1988.

[45] S. Javadi, M. Dahl, and M. I. Pettersson, "BTH trucks in aerial images dataset," *IEEE Dataport*, Dec. 2020, doi: 10.21227/qfpc-1s09.

SALEH JAVADI (Student Member, IEEE) received the B.Sc. degree in electrical-control engineering from Amirkabir University of Technology, in 2009, the M.Sc. degree in electrical, electronic, and systems engineering from the National University of Malaysia, in 2013, and the Licentiate degree in systems engineering from Blekinge Institute of Technology (BTH), Sweden, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and machine learning.

MATTIAS DAHL (Member, IEEE) received the B.Sc. degree in electrical engineering from Chalmers University of Technology, the M.Sc. degree in computer science from Lulea Institute of Technology, the Licentiate degree (engineering) in signal processing from Lund University, and the Ph.D. degree in applied signal processing from Blekinge Institute of Technology (BTH), Sweden, in 2000.

Since 2018, he has been a Full Professor with BTH. His research is focused on optimization of technical systems, self-learning methods, artificial intelligence, and analysis; it is conducted together with industry and the surrounding society, and several cases have resulted in patents. His current research is related to adaptive systems and processing, computer vision, automotive radar, and mathematical modeling.

Prof. Dahl received technology transfer scholarships from the Swedish Foundation of Technology Transfer and grants for verification for growth.

MATS I. PETTERSSON (Member, IEEE) received the M.Sc. degree in engineering physics, the Licentiate degree in radio and space science, and the Ph.D. degree in signal processing from Chalmers University of Technology, Gothenburg, Sweden, in 1993, 1995, and 2000, respectively.

For some years, he was in mobile communication research with Ericsson, and for ten years, he was also with Swedish Defence Research Agency (FOI). At FOI, he focused on Ultra-Wide Band low-frequency SAR systems. Since 2005, he has been with Blekinge Institute of Technology (BTH), where he is currently a Full Professor. His research is related to surveillance and remote sensing and his main interests are SAR processing, space-time adaptive processing (STAP), high resolution SAR change detection, automotive radar, radio occultation, and computer vision.