IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 15, 2020, accepted June 22, 2020, date of publication June 29, 2020, date of current version July 8, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3005391*

# Real-Time Fuel Truck Detection Algorithm Based on Deep Convolutional Neural Network

**HAMID R. ALSANAD[1,2], OSMAN N. UCAN[3], MUHAMMAD ILYAS[3], ATTA UR REHMAN KHAN[4], (Senior Member, IEEE), AND OGUZ BAYAT[1]**

[1]Institute of Graduate Studies, Altinbas University, 34217 Istanbul, Turkey
[2]Ministry of Science and Technology (MOST), Baghdad 10070, Iraq
[3]School of Engineering and Natural Sciences, Altinbas University, 34217 Istanbul, Turkey
[4]Faculty of Computing and Information Technology, Sohar University, Sohar 311, Oman

Corresponding author: Hamid R. Alsanad (hamidsend@yahoo.com)

**ABSTRACT** This paper presents a new approach by training and improving a convolutional neural network (CNN) based on You Only Look Once version 2 (YOLOv2) to efficiently detect fuel trucks from images in embedded systems. The proposed method considers the entire image area for strong object detection compared with existing methods that only focus on the image area where the class object exists to predict its probability to be in a class. The loss function for CNN is improved to enhance effective learning, especially when only a limited amount of data is available for training. The class probability can be learned by improving the loss function although the anchor boxes are not in the center of the target object. The learning process of the model can be in a limited range and achieve rapid convergence although the sizes of the initial anchor and target boundary boxes are different. Experimental results of various fuel truck images show the efficiency of the proposed approach under different detection scenarios of real fuel trucks. The detection rate of the proposed method is approximately 4% higher than the YOLOv2 object detection method. The proposed method is suitable to monitor long country borders using unmanned drones.

**INDEX TERMS** Convolutional neural network, CNN, object detection, fuel trucks, you only look once, YOLOv2.

## I. INTRODUCTION

Fuel is an important resource used by countries in the world. Thus, governments must take precautions to protect this wealth from smuggling. Smuggling gangs use many methods, such as using sea routes, small vessels or boats, and through land using different types of large vehicles, such as fuel trucks and others, to smuggle these resources outside the country borders. Many methods for antifuel smuggling are found in the literature, where object detection is considered a reliable method. Different object detection methods for persons, vehicles, and general things have been proposed. However, most of them focus on obtaining high detection accuracy while reducing high computational complexity. Several of those methods are unsuitable for real-time applications [1], [2].

Objects can be normally detected either from pictures or video feeds [3]. To detect an object, a bounding box is drawn around the area where it appears. Such a process involves two main steps, namely, classification of the object's type and drawing a box around that object. The width, height, and position of the object within the image are labeled and used as ground truth information in the deep network learning phase. Different types of object detection algorithms, such as region-based convolutional neural network (R-CNN), fast R-CNN, You Only Look Once (YOLO), YOLOv2, and YOLOv3, are used. The operation of these algorithms experiences many problems that affect the detection performance of the system. R-CNN requires a multistage learning process that results in slow object detection [4]. Fast R-CNN treats and improves this problem by simultaneously receiving and processing object candidate regions [5]. However, a considerable amount of time is consumed in calculating the candidate regions for object detection. The computation time using faster R-CNN is reduced by adding a network for finding candidate regions in the network for object classification [6]. The above methods require high computational throughput on graphics processing unit (GPU), thereby resulting in additional cost to the system and making them difficult to be used in real-time systems with limited computing power, such as embedded systems.

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

YOLO and its improved YOLOv2 are algorithms that can recognize various objects with high computational speed and identify their positions in the image [7], [8]. The learning in YOLOv2 is only performed for the cell where the object exists. Therefore, the learning of other neural network parts, that is, the area not belonging to the class, is not performed. Thus, a situation where a high-class probability is generated in an area where an object does not exist, and the training efficiency is lowered because only the anchor boxes belonging to the cell where the center of the object is located in the training image are used for class probability learning. Therefore, the efficiency of the YOLOv2 algorithm is low.

This paper mainly aims to modify the operation mechanism of YOLOv2 for improving its object detection efficiency by introducing a deep network-based algorithm and its learning method for efficient object detection. The proposed method improves the loss function compared with that of YOLOv2. Thus, the class probability can be learned, especially on anchor boxes that do not belong to the center of the object. This model is designed to limit the range and to quickly converge although the sizes of the initial anchor and target bounding boxes are different in the learning process. Intersection over Union (IOU) between the target bounding and anchor boxes is used to enable stable convergence of center position estimation and bounding box estimation of the object. After network adjustment toward the targeted object detection, this classification network provides an overall increase of approximately 4% mean average precisions (mAPs).

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 proposes a different version of YOLOv2, which is called as the optimized YOLOv2 fuel truck detection (OYOLOv2_FTD). Section 4 discusses the datasets and experimental results, and Section 5 provides the conclusion and future work.

## II. RELATED WORK

Several studies have analyzed and evaluated object detection performance using various algorithms. Recent relevant studies are reviewed below:

Viola and Jones [9] used haar features and proposed an important task of the object detection framework. They accomplished object detection using cascade classifiers and Adaboost training with a technique called the sliding window. Dalal and Triggs [10] performed classification on subwindows extracted from the image itself using a linear support vector machine (SVM) with a sliding window mechanism. They introduced an efficient histogram of oriented gradient (HOG) features based on the directions of the edge in an image for detection. Felzenszwalb *et al.* [11] proposed a graphical model called deformable part model (DPM) to overcome the distortions of objects within the images. DPM technology assumes that an object is composed of different parts. DPM utilizes linear SVM and HOG combined with the sliding window technique. Regionlets [12] enhanced the DPM by searching the possible positions at the different portions of the object. Pepik *et al.* [13] expanded the DPM into 3D object formation.

Uijlings *et al.* [14] suggested selective search (SS) to produce a group of data-driven, category-independent object suggestions and avert the use of a conventional sliding window search. The operation of SS depends on hierarchical segmentation applying various groups of clues. They used SS to generate a localization and recognition system model based on a bag-of-words. LeCun *et al.* [15] and Krizhevsky *et al.* [16] illustrated that the remarkable success of the CNN as an ideal extractor of features for image recognition and classification has a considerable effect on object detection. The top object detectors utilizing CNNs can be summarized as follows:

Sermanet *et al.* [17] proposed a feature extractor called the OverFeat framework. CNN was first utilized using the sliding window technique. Girshick *et al.* [4] and Uijlings *et al.* [14] proposed a region-based CNN (R-CNN) that requires a multistage learning process (SS is used to generate proposals of candidate regions for the input image, the features of each proposal are calculated, a linear SVM is applied to classify each region, the bounding boxes are fine-tuned, and redundant ones are removed in postprocessing). These processes result in slow object detection. Fast R-CNN [5] improves object detection using spatial pyramid pooling networks [18] to simultaneously receive and process object candidate regions. Faster R-CNN [6] reduces the computation time by adding a network-based region proposal [19], which is a kind of fully connected network [20], for finding candidate regions.

Xiang *et al.* [21] modified fast R-CNN [5] to improve the object candidate generation stage by entering subcategory information for 3D voxel patterns [22] to the network. Cai *et al.* [23] detected multiple layers of the network to handle objects of different sizes. Chen *et al.* [24] developed 3D proposals, these proposals are recorded by several contextual and segmentation features for 3D object detection, and the rescoring utilizes the fast R-CNN [5]. Yang *et al.* [25] introduced a method depends on rejecting negative proposals of the object using cascaded classifiers and convolutional features. A scale-dependent object classifier was used to evaluate the remaining proposals.

References [7], [8] and [26] considered the detection of objects as a regression problem using YOLO and single-shot detector (SSD) [27] algorithms and removed the stage of object proposal creation. These algorithms are based on an individual CNN shadowed by a nonmaximal suppression stage. In those algorithms, all images that enter the network are separated into (7 × 7 grid, 13 × 13 in YOLO, and 9 × 9 in SSD), where the prediction of a fixed number of boundary boxes is the responsibility of each grid cell. Their major advantages are the construction of a fast detector and viewing the entire image through training, thereby increasing the accuracy of detection. Although SSD is a modern object detection structure, hard negative mining is difficult, and sampling selection results in high confidence loss. The two major drawbacks of these methods are 1) imposing strict limitations on the prediction of bounding box (for example, every grid cell in YOLO can only predict two boundary boxes), and 2) detection of small objects is extremely difficult. The SSD system attempted to fix the second drawback by increasing the datasets for small objects. Fu *et al.* [28] propose a deconvolutional SSD, which is the improved version of SSD, to focus on the context with the help of deconvolutional layers.

In reference [29], YOLOv2 is used to detect vehicles in unmanned aerial vehicle images, which is the beginning of successful use CNN regression-based in the city administration. References [30]–[35] explored vehicle detection using CNNs. Such methods do not need human-involved features, and only a large number of tagged vehicle images are used to train the network with supervision before the network can automatically learn the vehicle-type features.

The main objectives of YOLOv2 [8], which is the basis of the proposed method and is the second version of YOLO, are to significantly improve the accuracy while reducing the complexity, that is, fast performance. YOLOv2, which is an optimized model of YOLOv1, maintains the speed advantages and increases the mAP rate of YOLOv1 (63.4). YOLOv2 can be operated at varying and different sizes, thereby providing an easy trade-off between accuracy and speed using a new multi-scale training method. A list of significant solutions was added in YOLOv2 increasing the mAP. Batch normalization is used for pre-processing the data of input image. The high-resolution classifier of YOLOv1 at 224 × 224 to 448 × 448 increases the mAP by 4%. A modern network with the ''network in network'' concept [36] used by YOLOv2 expects with comprehensive average pooling and pads the features by inserting 1 × 1 convolutional core among the 3 × 3 convolutional cores.

The YOLOv2 network has few convolutional layers (19 compared with 24 in YOLOv1) and few filters plus five max pooling layers, making it more reliable than its predecessor. YOLOv2 supports convolution with anchor boxes, increases the resolution of each grid from 7 × 7 in YOLOv1 to 13 × 13, and only has a single bounding box per grid. After examining the curve of IOU function by calculating the statistics of ground truth on the image sets (Visual Object Classes (VOC) and Common Objects in Context) through the use of the k-means algorithm, YOLOv2 detects a solution for the best amount of anchor boxes and determines the anchor boxes number to be five, where its average IOU (61.0) is similar to faster R-CNN (60.9) that uses nine anchor boxes. The location is immediately predicted to overcome the instabilities caused by anchor boxes, thereby leading to a 4% mAP increase with the positive effect of dimension clusters. To obtain the features extracted from the previous 26 × 26 layer, YOLOv2 subjoins a pass-through layer and combines it with the features of the original final output to enhance the facility of detecting small objects. Thus, YOLOv2 increases the mAP by 1%.

## III. PROPOSED ALGORITHM

The proposed algorithm called OYOLOv2_FTD focuses on improving YOLOv2 in detecting fuel trucks. This method uses a single deep neural network to simultaneously predict the location of fuel trucks and classify their category. The proposed algorithm adopts three approaches, namely,

1. Reducing the number of CNN layers.
2. Reducing the number of anchors where the amount of processing and the required time are reduced.
3. Modifying the loss function.

The details of the network structure and learning process to predict these outputs are described in the following subsections.

### A. PROPOSED NETWORK STRUCTURE

A YOLOv2-based [8] end-to-end training CNN is proposed to detect fuel trucks. The proposed deep network structure is shown in Figure **1**, and its layers are described in Table 1. This structure is a simplified structure of YOLOv2. Each layer is composed of a convolutional layer and a maxpool layer. A leaky rectified linear activation function is used.

The proposed OYOLOv2_FTD structure uses only one anchor box compared with the existing YOLOv2 that uses five anchor boxes. An anchor box size and position estimation method is proposed to compensate for this difference.

The fuel trucks are estimated using the deep network, the object class is set to one, and the image is divided into 13 × 13 grids, similar to YOLOv2, to estimate the position and

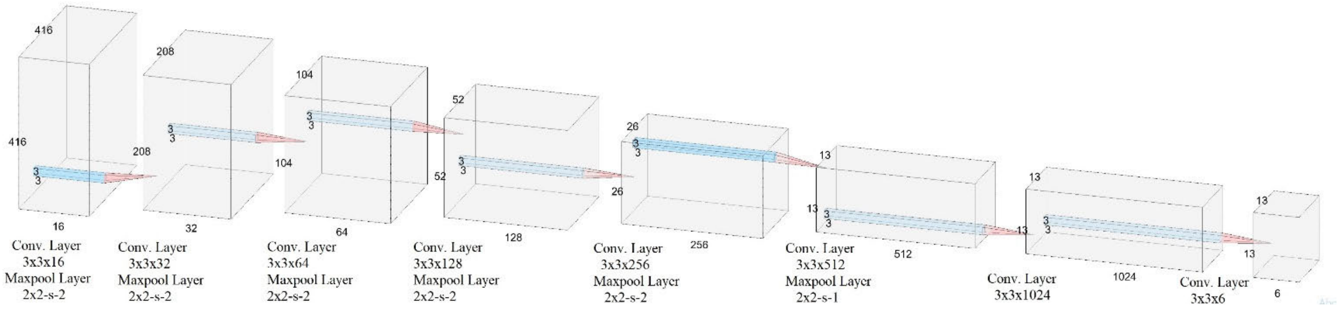

FIGURE 1. Diagram of the proposed CNN structure.

**TABLE 1.** Description of the proposed CNN.

| Layer | Type | Size/Stride | Filters | Output |
|---|---|---|---|---|
| *1* | Conv<br>Max-pool | (3×3)<br>(2×2)/2 | 16 | 416 × 416 × 16 |
| *2* | Conv<br>Max-pool | (3×3)<br>(2×2)/2 | 32 | 208 × 208 × 32 |
| *3* | Conv<br>Max-pool | (3×3)<br>(2×2)/2 | 64 | 104 × 104 × 64 |
| *4* | Conv<br>Max-pool | (3×3)<br>(2×2)/2 | 128 | 52 × 52 × 128 |
| *5* | Conv<br>Max-pool | (3×3)<br>(2×2)/2 | 256 | 26 × 26 × 256 |
| *6* | Conv<br>Max-pool | (3×3)<br>(2×2)/1 | 512 | 13 × 13 × 512 |
| *7* | Conv | (3×3) | 1024 | 13 × 13 × 1024 |
| *8* | Conv | (3×3) | 6 | 13 × 13 × 6 |

size of fuel trucks using the anchor boxes belonging to each grid cell. Thus, the final output of the deep network is 13 × 13 grid cells, with one anchor box for each cell, and six outputs for each anchor box (2D center coordinates (x, y), width and height (h, w), one class probability, one confidence). Thus, the deep network has an output size of 13 × 13 × 6, as shown in layer 8 of Table 1.

Although this paper deals with one class, the number of classes can be extended to multiple classes when the filter size is modified to match the number of classes in layer 8. However, the depth of the deep network should be increased because it has eight layers for effective object detection, which is relatively shallow compared with other deep network structures. Although this network is relatively shallow, sufficient detection is possible with low hardware complexity.

## B. CLASS PROBABILITY AND CONFIDENCE LEARNING

As previously mentioned, the outputs of the deep network are the six values of the anchor box belonging to each cell of the 13 × 13 grid cells, which are the coordinates of the center, the width and height of the box, class probability P (class|object), and confidence

(C = P(object) IOU). Confidence is the probability of object existence P(object) multiplied by IOU. To learn the class probability, object existence probability, and confidence in YOLOv2, we set the value of P(class|object) = P(object) = 1 in the cell where the center of the class object belongs in the training data, and we set P(object) = 0 for the remaining cells. The class probability is only trained for cells with probability 1, and the object existence probability is trained for all cells. In this case, class probability learning is not performed for cells with P(class|object) = 0. The estimation of an object with high existence probability but does not belong to any class and the image region where an object does not exist recognized as an object caused by false positive are not considered. YOLOv2 determines that the class with the highest probability within the class defined by a softmax function as the final class of the object, although the image area recognized as the real object does not belong to the class defined in the deep network. In particular, no class comparison target is found when only one class needs to be estimated. Thus, the class probability becomes one in all cases using the softmax function, and the deep network cannot perform the classifier function.

In this paper, a sigmoid function is first applied before applying the softmax function in calculating the class probability, and the result is higher than the threshold value. The softmax function is then applied to calculate the final class function. The class probability is trained to make it converge to P(class|object) = 0 for the cell where the object's center belongs and for the other cells. The deep network is converted to the estimated class probability value and confidence through Equations (1a & 1b), similarly to YOLOv2, to estimate the target class probability value and confidence.

$$\hat{P}\,(class \,|\, object) = sig\,(t_{class})\,, \tag{1a}$$

$$\hat{C} = sig\left(t_{Object}\right). \tag{1b}$$

where $t_{class}$ and $t_{object}$ are the outputs of the deep network to estimate the class probability and confidence, respectively.

## C. TRAINING OYOLOv2_FTD TO ESTIMATE THE COORDINATES AND SIZE OF ANCHOR BOXES

This subsection focuses on training OYOLOv2_FTD to estimate the coordinates and size of anchor boxes. The center coordinates of the anchor box are shown in Figure **2**. The offset is defined using the vertex at the upper-left corner of the cell where the center of the object is located, similar to YOLOv2. The x and y offsets range are from 0 to 1 because the side length of each cell is 1. Therefore, we estimate the offset using the sigmoid function. For YOLOv2, the estimated width $\hat{w}$ and height $\hat{h}$ of the anchor box are calculated using Equation (2).

$$\hat{w} = P_{width} exp\,(t_{width})\,, \hat{h} = P_{height} exp\left(t_{height}\right), \tag{2}$$

where $t_{width}$ and $t_{height}$ are the output values for estimating the width and height of the anchor box in the deep network, and $P_{width}$ and $P_{height}$ represent the initial estimated width and

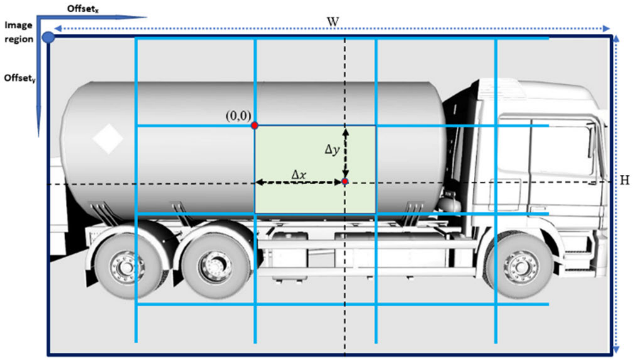

**FIGURE 2.** x and y offsets of a bounding box are defined from the top-left corner of the grid cell where the center of the object is located.

height of the anchor box. Since the image is converted to the size W×H at the output of the deep network, the range of the estimated width and height in Equation (2) should be limited to $0 \leq \hat{w} \leq$ W and $0 \leq \hat{h} \leq$ H, respectively.

The above equation does not satisfy this formula, and because of the characteristics of the exp function, the rate of change when $t_{width}$ and $t_{height}$ has a negative value is smaller than the rate of change when having a positive value. The disadvantage is that the convergence time increases when the width and height values are relatively larger than the estimated width and height values of the actual object. Equations (3a–d) are proposed in OYOLOv2_FTD to address this drawback.

$$\hat{w} = Wsig\left(t_{width} - a_{width}\right), \tag{3a}$$

$$\hat{h} = Hsig\left(t_{height} - a_{height}\right), \tag{3b}$$

where:

$$a_{width} = log\left(\frac{W - P_{width}}{P_{width}}\right), \tag{3c}$$

$$a_{height} = log\left(\frac{H - P_{height}}{P_{height}}\right). \tag{3d}$$

Through the above equations, the range of the estimated width and the estimated height can be limited to $0 \leq \hat{w} \leq W$ and $0 \leq \hat{h} \leq H$, respectively. Also, due to the characteristics of the sigmoid function $P_{width}$ and $P_{height}$ are based on W/2 and H/2. They have different rates of change, that can quickly converge although the difference between the initial setting values $P_{width}$ and $P_{height}$ and the width and height of the target bounding box is large.

### D. SETTING LOSSES FOR OPTIMIZATION LEARNING

We use the adaptive gradient (AdaGrad) algorithm, belonging to the gradient descent-based learning method for learning the deep network. As illustrated in Equations (4–9) the loss function ($L_{total}$) that the algorithm aims to minimize is defined as the weighted sum of the following:

- The localization loss ($L_{centercoord}$): It involves two parts of losses, the first part is the loss from bounding box coordinate x and y ($L_{xy}$) and the second part is the loss from width w and height h ($L_{wh}$).
- The classification loss ($L_{class}$): It is the class probability loss of grid cell.
- The confidence loss ($L_{confidence}$): It is the loss from the confidence in each bound box.

$$L_{total} = (W_{centercoord}L_{centercoord} + W_{class}L_{class} + W_{confidence}L_{confidence}), \tag{4}$$

where:

$$L_{centercoord} = \left(L_{xy} + L_{wh}\right), \tag{5}$$

$$L_{xy} = \sum_{i=0}^{s^2} I_i^{obj} \exp\left(-IOU_i\right)\left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2\right], \tag{6}$$

$$L_{wh} = \sum_{i}^{s^2} I_i^{obj} \exp\left(-IOU_i\right) \times \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right], \tag{7}$$

$$L_{class} = \sum_{i}^{S^2} \sum_{c\in class} [\lambda_{class}^{obj} I_i^{obj}\left(1 - \hat{P}_i\left(class\,|\,object\right)\right)^2 + \lambda_{class}^{noob} I_i^{obj} \hat{P}_i\left(class\,|\,object\right)^2], \tag{8}$$

$$L_{confidence} = \sum_{i}^{S^2}\left[\lambda_{confidence}^{obj} I_i^{obj}\left(1 - \hat{C}_i\right)^2 + \lambda_{confidence}^{noob} I_i^{noob} \hat{C}_i^2\right], \tag{9}$$

where $L_{xy}$, $L_{wh}$, $L_{confidence}$, and $L_{class}$ are the sums of square errors of the center coordinates, width and height, confidence, and class probability between the target bounding box and the anchor box, respectively. $I_i^{obj}$ has a value of 1 when the center of the bounding box is located in the i-th cell, otherwise, it is 0, and $I_i^{noob}$ has the opposite value. In the formulas of $L_{xy}$ and $L_{wh}$, $IOU_i$ denotes to the IOU between the anchor box and the target bounding box belonging to the i-th cell. As learning progresses, the overall loss decreases, and the IOU value of the anchor box belonging to the cell where the center of the target bounding box is located decreases. In this paper, the weight of $L_{centercoord}$ in Equation (4) is applied by applying exp(–$IOU_i$) to the internal weight of $L_{xy}$ and $L_{wh}$. Although the internal weight value of $L_{centercoord}$ itself is large, it can be reduced with the decrease in loss.

For $L_{class}$, we learn that the class probability for the anchor box belonging to the cell where the object does not exist is zero and separate the confidence and class probability by learning the probability of $P_i$(class|object) rather than $P_i$(class), indicating that each class probability can be learned. $L_{confidence}$ is configured to learn all cells similar to the loss function of YOLOv2. For $L_{total}$, the specific gravity of each loss, that is, the $W_{class}$, $W_{centercoord}$ and $W_{confidence}$, values can be adjusted by the characteristics of the object to be detected because the scale between each loss value is different.

For the images of fuel trucks captured with a camera placed on drones, the size proportion occupied by the image is small. Thus, the values of $L_{confidence}$ and $L_{class}$ that consider all cells are larger than $L_{xy}$ and $L_{wh}$ that only consider cells where their class probability is not zero. On this basis, the relative scale between $L_{confidence}$ and $L_{class}$ can be adjusted by setting $W_{centercoord}$ to a value that is relatively large compared with $W_{confidence}$ and $W_{class}$. For $L_{confidence}$ and $L_{class}$, the class probability for small objects is small, and the relative importance can be controlled by setting $\lambda_{confidence}^{obj}$ and $\lambda_{class}^{obj}$ with values larger than $\lambda_{confidence}^{noob}$ and $\lambda_{class}^{noob}$.

## IV. DATASETS AND EXPERIMENTAL RESULTS

The main objective of this paper is to modify the operation mechanism of YOLOv2 to improve its object detection efficiency. The modified YOLOv2 is tested on fuel trucks as a case study. This section describes the datasets used in the experiments, the evaluation measures, and the results of the conducted experiments.

### A. DATASETS

Publicly available datasets, including PASCAL VOC 2007 and PASCAL VOC 2012 datasets, are used to investigate the performance of the proposed method [37]–[39]. We also build our own dataset of fuel trucks that contains 2500 fuel truck images in different environments and orientations. The images in the datasets include daily scenes on sunny and cloudy days, and noise background, people, snow, rain, and other vehicle types are found. This dataset is divided into training and test datasets with the desired ratio, such as 6:4 or 7:3. We use 7:3 in this paper, as shown in Table 2.

**TABLE 2. Fuel trucks dataset distribution.**

| Datasets | Number of Samples |
|---|---|
| *Training Set* | 1750 |
| *Testing set* | 750 |
| *Total* | 2500 |

For the object detection task, the boundary and label for each object's ground truth must be manually specified. Standardized ground truth marking methods are provided in the PASCAL VOC dataset. This method is also used in the fuel truck dataset to create the bounding boxes and labels.

The Image Labeler application on MATLAB, which is a widely adopted annotation tool in numerous applications, is used to annotate the fuel truck images. This tool directly converts the annotation message into a mat file format containing the image's number and class(es) with boundary boxes. Image annotation is manually conducted for the 2500 fuel truck images. The schematic of the annotated fuel trucks is shown in Figure **3**. The annotated image is used as an input to train the proposed OYOLOv2_FTD architecture.

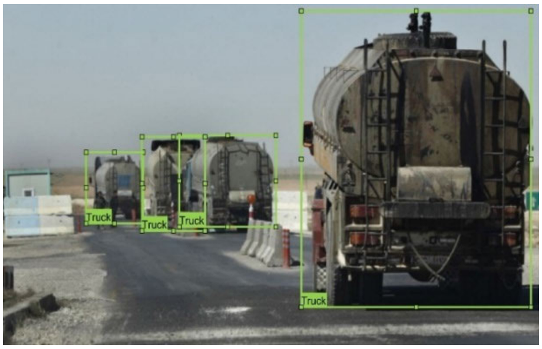

**FIGURE 3. Sample labeling of fuel trucks using the Image Labeler application.**

### B. EVALUATION METHODS

The proposed OYOLOv2_FTD algorithm is trained on the fuel truck dataset and PASCAL VOC2007 and PASCAL VOC2012 datasets and compared with the existing YOLOv2 algorithm to verify its performance. The metrics used to evaluate the proposed model are as follows: IOU, accuracy, and mAP. IOU denotes the overlap rate of the predicted bounding box generated by the network and ground truth bounding box. The bounding box is considered to be correct when IOU overrides the threshold, as shown in Equation (10). This standard is used to measure the correlation between the object ground truth and object prediction, where the higher the correlation is, the greater the value will be. IOU is used to calculate the AP of the proposed fuel truck detection model.

The bounding box ($B_{pred}$) of a fuel truck in the image is collected from the neural network after dropping the input image. Equation (10) is satisfied when the IOU value is greater than the value of threshold $a_0$, and the prediction is regarded as correct based on the IOU value of the predicted bounding box of fuel trucks ($B_{pred}$) and ground truth bounding box ($B_{truth}$). Figure **4** shows the detected fuel trucks and ground truth areas, where the areas marked in purple and blue represent the predicted bounding boxes, and the ground truth areas are marked in green. This process aims to calculate the IOU between the predicted and ground truth areas.

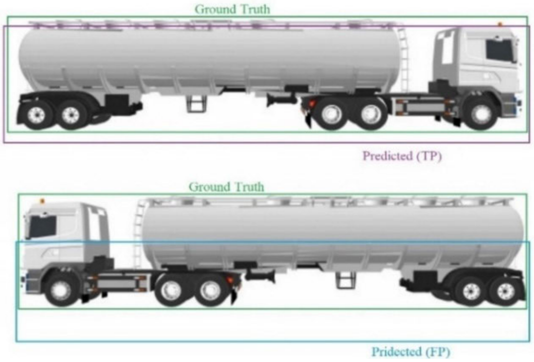

**FIGURE 4. A calculation example of IOU [40].**

The result is considered to be true positive (TP) when the IOU value is larger than 50% of the threshold, otherwise, it is considered as false positive (FP). False negative (FN) occurs when the model predicts no fuel trucks in the image, but the image contains fuel trucks. Accordingly, these conditions can be combined into two metrics, namely, precision and recall.

$$a = \frac{B_{pred} \cap B_{truth}}{B_{pred} \cup B_{truth}} \geq a_0, \tag{10}$$

The accuracy of the predictions provided by the object detector is measured using the IOU evaluation metric. The importance of this evaluation metric is by setting anchor boxes while preparing the training data and is significantly important when non-suppression is used for the cleanup required when multiple boxes are predicted per each object.

The value of $a_0$ is set to 0.5, indicating that the intersection between the predicted and ground truth regions is at least half of the size of the ground truth area. The test case is predicted as a fuel truck when the IOU value is larger than the 0.5 threshold.

Precision is defined as the ratio of TP to the total positive predictions, which can be expressed as:

$$Precision = \frac{TruePositives}{Alldetections} = \frac{TP}{TP+FP} = \frac{TP}{n}, \qquad (11)$$

where n (TPs + FPs) represents the total number of images recognized by the system.

$$Recall = \frac{TruePositives}{Alltrueinstances} = \frac{TP}{TP+FN}, \qquad (12)$$

Accuracy is an intuitive measure of performance and is the ratio of correctly predicted observation to the total observations. Accuracy is calculated as follows:

$$Accuracy = \frac{TP+FN}{TN+TP+FP+FN}, \qquad (13)$$

The denominator of recall is the sum of TPs and FNs. The summation of the two values can be recognized as the total number of fuel trucks in the ground truth. An additional evaluation measure, mAP, is equal to the area below the curves of recall and precision. The values of this measure are limited in the interval [0, 1], where large values indicate good detection accuracy.

### C. EXPERIENTIAL RESULTS

The proposed OYOLOv2_FTD model is trained using the fuel truck dataset. The images in this dataset have different sizes, resolutions, and different environments. This dataset is used to verify the robustness of the proposed model under different scenarios.

For a fair comparison, all experiments are conducted on a computer with Intel(R) Core (TM) i7-6700HQ CPU @ 2.60 GHz CPU and NVIDIA GeForce GTX 965M GPU, with 16 GB memory and CUDA10.1 cuDNN9.1. The operating system of the computer is a 64-bit Windows 10. We adopt a MathWorks support example (Object Detection Using YOLOv2 Deep Learning) [41] to train the proposed deep learning model. This example, which is already trained using VOC 2007+2012, is selected as the backbone of the proposed OYOLOv2_FTD. The results of the conducted experiments verify the effectiveness and accuracy of the proposed method in terms of computational cost.

YOLOv2 and OYOLOv2_FTD are separately trained on the same dataset to verify the performance of the proposed model. The loss values during the training phase of the two models are shown in Figure **5**. The x-axis denotes the number of iterations, and the y-axis denotes the loss values for each iteration. The losses of the two models decrease and converge to 0, when the iteration number increases, confirming that the proposed model is successfully trained with the training dataset. As shown in Figure **5**, the loss of OYOLOv2_FTD is lower and converges to zero faster than the original YOLOv2 model. The proposed model outperforms the YOLOv2 model.

The proposed algorithm immensely improves the average accuracy compared with the YOLOv2 algorithm using the fuel truck dataset. The proposed OYOLOv2_FTD algorithm achieves 93.0 mAP, and the original YOLOv2 achieves 89.0 mAP, as shown in Table 3.

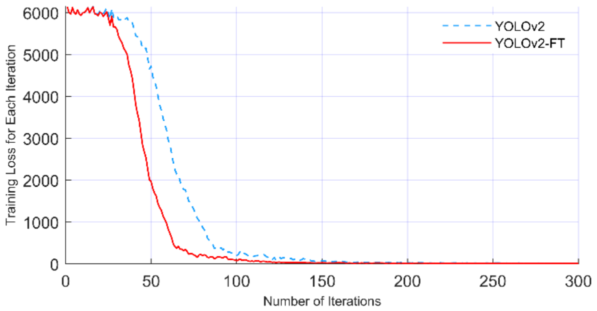

**FIGURE 5.** Loss curves of the two models.

**TABLE 3.** Fuel trucks test detection results of the two Algorithms.

| Algorithms | Dataset | mAP |
|---|---|---|
| *YOLOv2* | Fuel truck | 89 .0 |
| *OYOLOv2_FTD* | Fuel truck | 93.0 |

We train the proposed OYOLOv2_FTD algorithm for detection using the VOC2007 dataset. Table 4 shows the performance comparison of selected (less than real-time) detectors like Fastest DPM [42], R-CNN Minus R [43], Fast R-CNN [5], and the proposed OYOLOv2_FTD algorithm. As shown in the results, OYOLOv2_FTD method outperforms all other methods. The proposed OYOLOv2_FTD algorithm achieves 80.9 mAP, while the Fast R-CNN, R-CNN Minus R, Fastest DPM achieve 70.0 mAP, 53.5 mAP, 30.4 mAP respectively.

**TABLE 4.** PASCAL voc2007 test detection results of different less than real-time models.

| Algorithms | Dataset | mAP |
|---|---|---|
| *Fastest DPM* [42] | VOC2007 | 30.4 |
| *R-CNN Minus R* [43] | VOC2007 | 53.5 |
| *Fast R-CNN* [5] | VOC2007 | 70.0 |
| *OYOLOv2_FTD* | VOC2007 | 80.9 |

We also trained the proposed algorithm using the VOC (2007+2012) datasets and compared its results with some (real-time) detectors like Fast YOLO [7], YOLO(YOLOv1) [7], SSD [27], YOLOv2 [8]. The result is shown in Table 5.

Figure **6** illustrates some results showing that the system successfully detects fuel trucks in different scenes and under different backgrounds.

Figure **7** shows the TP, FP, and FN rates of the two algorithms. The average of the results of the two algorithms is shown in a pie graph. As shown in the ratios on the pie graph, the proposed algorithm learns the class probabilities of regions in the image where no fuel trucks are present during training, indicating high TP and low FP and FN rates. For the detection speed per image, YOLOv2 has an average time

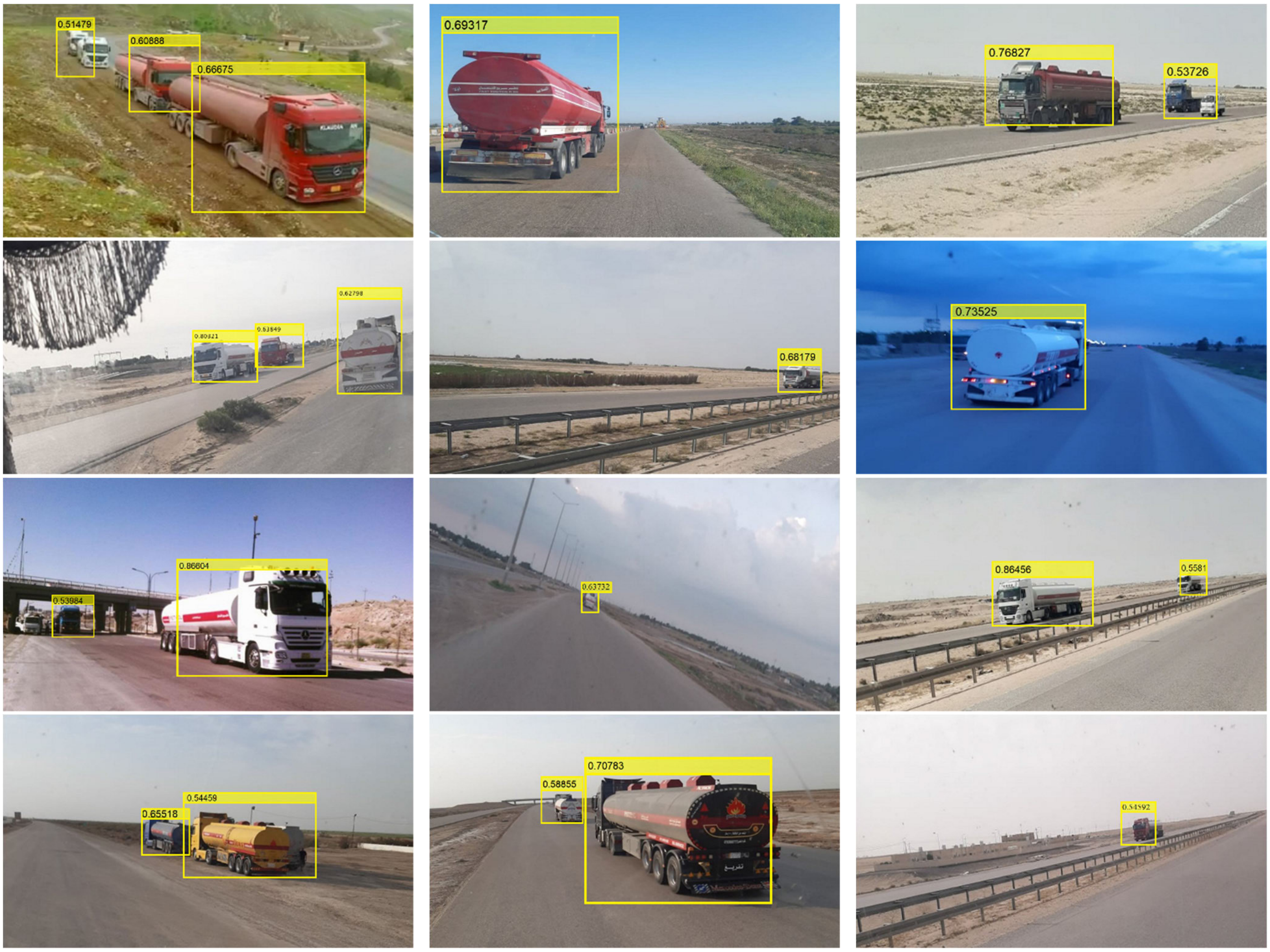

**FIGURE 6.** Selected examples of fuel truck detection results on the fuel truck dataset using the proposed OYOLOv2_FTD. Each output box is associated with the score, and the detected frame rate is displayed in the upper-left corner.

**TABLE 5.** PASCAL VOC2012 test detection results of different real-time models.

| Algorithms | Dataset | mAP |
|---|---|---|
| *Fast YOLO* [7] | VOC (2007+2012) | 52.7 |
| *YOLO(YOLOv1)* [7] | VOC (2007+2012) | 63.4 |
| *SSD* [27] | VOC (2007+2012) | 74.3 |
| *YOLOv2* [8] | VOC (2007+2012) | 78.6 |
| *OYOLOv2_FTD* | VOC (2007+2012) | 82.1 |

of 0.0416 s, and the proposed algorithm has an average time of 0.0191 s. This finding is because YOLOv2 uses five anchor boxes, and the proposed algorithm detects fuel trucks using one anchor box to improving the size estimation of the anchor box, as shown in Equations (3a-3d).

As shown in Figure **8**, the proposed OYOLOv2_FTD has an AP value of 93% on the testing set, which is higher than the YOLOv2 method (89%).

## V. OYOLOv2_FTD LIMITATIONS

The main limitation of our proposed model (OYOLOv2_FTD) as compared to the original model (YOLOv2) is that the operation of our model is based on using one anchor box rather than five. Consequently, if two or more fuel trucks lay in the same cell (a very rare scenario), it will be considered as one object. However, it is noteworthy that using one anchor box provides faster training and detection compared to the original model. The second limitation of this work is its comparison with YOLOv3. We selected YOLOv2 for improvement because YOLOv3 has worst performance for detecting large objections [26]. It would be good that the proposed model must be compared with YOLOv3, but YOLOv3

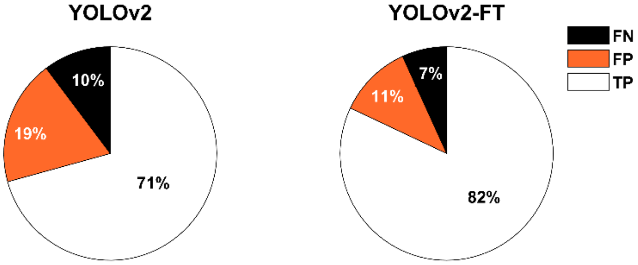

**FIGURE 7.** TP, FP, and FN rates of YOLOv2 and the proposed OYOLOv2_FTD algorithm during detection.

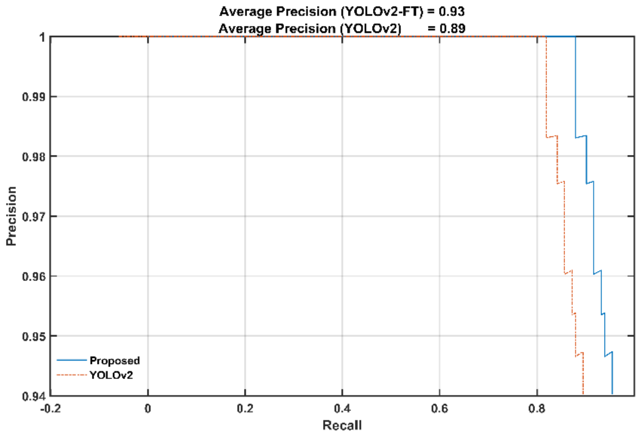

**FIGURE 8.** Precision recall curve performance of the proposed OYOLOv2_FTD detection and YOLOv2 detection.

has not been trained on PASCAL VOC (2007 and 2012). Considering the excessive work, this comparison is beyond the resources of this project.

## VI. CONCLUSION

This paper focuses on the detection of fuel trucks based on a deep neural network called YOLOv2. YOLOv2 is used because of its ability to outperform current methods in terms of accuracy and performance in near real-time. We proposed a new network based on YOLOv2 method, which is called the real-time fuel truck detection algorithm based on deep CNN. The proposed OYOLOv2_FTD method improves the class probability and regression learning of the coordinate system in the existing YOLOv2 method, where it can be efficiently trained using a small amount of training data. The class probability can be learned by improving the loss function although the anchor boxes are not in the center of the target object. The learning process of the model can be in a limited range and achieve rapid convergence although the sizes of initial anchor and target boundary boxes are different. The results of the conducted experiments show that the mAP of the proposed OYOLOv2_FTD detection model can reach 93.0%. The proposed OYOLOv2_FTD algorithm outperforms the traditional YOLOv2 algorithm by 4% in detecting fuel trucks. Although the proposed model achieves good performance in the conducted experiments, the number of fuel trucks and the amount of data is relatively low. In future research, we will collect many actual fuel truck data to improve the accuracy and speed of fuel truck detection. The proposed algorithm is suitable for detecting and tracking fuel trucks in real-time.

## REFERENCES

[1] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “DeepDriving: Learning affordance for direct perception in autonomous driving,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.

[2] J. E. Hoo and K. C. Lim, “Accuracy and error study of horizontal and vertical measurements with single view metrology for road surveying,” *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 12, pp. 7872–7876, 2016.

[3] J. C. Nascimento and J. S. Marques, “Performance evaluation of object detection algorithms for video surveillance,” *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 761–774, Aug. 2006.

[4] 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.

[5] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[6] 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.

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

[8] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7263–7271.

[9] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, vol. 1, nos. 511–518, p. 3.

[10] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[12] X. Wang, M. Yang, S. Zhu, and Y. Lin, “Regionlets for generic object detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2071–2084, Oct. 2015.

[13] B. Pepik, M. Stark, P. Gehler, and B. Schiele, “Multi-view and 3D deformable part models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2232–2245, Nov. 2015.

[14] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” Feb. 2014, pp. 1–15, *arXiv:1312.6229v4*. [Online]. Available: https://arxiv.org/abs/1312.6229

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[21] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, “Subcategory-aware convolutional neural networks for object proposals and detection,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 924–933.

[22] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, “Data-driven 3D voxel patterns for object category recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1903–1911.

[23] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 354–370.

[24] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3D object detection for autonomous driving,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[25] F. Yang, W. Choi, and Y. Lin, “Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.

[26] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. Comput. Vis. Pattern Recognit. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[28] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

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

[30] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, Dec. 2018.

[31] O. Acatay, L. Sommer, A. Schumann, and J. Beyerer, "Comprehensive evaluation of deep learning based detection methods for vehicle detection in aerial imagery," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.

[32] M. Feng, S. Hu, G. Lee, and M. Ang, "Towards precise vehicle-free point cloud mapping: An on-vehicle system with deep vehicle detection and tracking," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 1288–1293.

[33] Z. Xu, H. Shi, N. Li, C. Xiang, and H. Zhou, "Vehicle detection under UAV based on optimal dense YOLO method," in *Proc. 5th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2018, pp. 407–411.

[34] B. Arsenali, P. Viswanath, and J. Novosel, "RotInvMTL: Rotation invariant MultiNet on fisheye images for autonomous driving applications," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2373–2382.

[35] J. Xu, Y. Nie, P. Wang, and A. M. Lopez, "Training a binary weight object detector by knowledge transfer for autonomous driving," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2379–2384.

[36] M. Lin, Q. Chen, and S. Yan, "Network in network," Mar. 2014, pp. 1–10, *arXiv:1312.4400v3*. [Online]. Available: https://arxiv.org/abs/1312.4400

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2012 (VOC2012) development kit," Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep., 2012, vol. 2007, pp. 1–45.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[40] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, "Ship detection based on YOLOv2 for SAR imagery," *Remote Sens.*, vol. 11, no. 7, p. 786, Apr. 2019.

[41] MathWorks. (2019). *Object Detection Using YOLO v2 Deep Learning*. Accessed: Sep. 14, 2019. [Online]. Available: https://www.mathworks.com/help/vision/ug/train-an-object-detector-using-you-only-look-once.html?searchHighlight=ObjectDetectionUsingYOLOv2DeepLearning&s_tid=doc_srchtitle

[42] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2497–2504.

[43] K. Lenc and A. Vedaldi, "R-CNN minus R.," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, X. Xie, M. W. Jones, and G. K. L. Tam, Eds. Dulles, VA, USA: BMVA Press, Sep. 2015, pp. 5.1–5.12.

**HAMID R. ALSANAD** received the B.S. and M.S. degrees in electrical and electronic engineering from the Al-Rasheed College of Engineering and Science, University of Technology, Baghdad, Iraq, in 1991 and 2005, respectively. He is currently pursuing the Ph.D. degree with Altinbas University, Istanbul, Turkey. His research interests include computer vision and mobile ad hoc networks.

**OSMAN N. UCAN** received the B.S., M.S., and Ph.D. degrees from Istanbul Technical University, Istanbul, Turkey, in 1985, 1988, and 1995, respectively. He is currently a Lecturer (Professor) and the Dean of the School of Science and Engineering, Altinbas University. He has more than 300 publications. His current research interests are in the areas of wireless communications, digital signal processing, channel coding, channel estimation, and equalization techniques, in particular multilevel coding, turbo coding, and equalization for wireless communication systems.

**MUHAMMAD ILYAS** received the B.S. degree in computer engineering from COMSATS IIT Abbottabad, Pakistan, in 2010, and the M.S. and Ph.D. degrees in electrical and computer engineering from Altinbas University, Istanbul, Turkey, in 2014 and 2018, respectively. He is currently a Lecturer (Assistant Professor) with Altinbas University. His current research interests include *in vivo* wireless communications, wireless body area networks, neural networks, and implantable devices.

**ATTA UR REHMAN KHAN** (Senior Member, IEEE) was the Director of the National Cybercrime Forensics Lab Pakistan and the Head of the Air University Cybersecurity Center. He is currently an Associate Professor with the Faculty of Computing and Information Technology, Sohar University, Oman. His research interests include cybersecurity, mobile cloud computing, ad hoc networks, and the IoT. He has received multiple awards, fellowships, and research grants. He is a Steering Committee Member/Track Chair/Technical Program Committee (TPC) Member of over 70 international conferences. He also serves as a domain expert for multiple international research funding bodies. He was the Conferences Chair of the IEEE Islamabad Section. He is currently serving as an Associate Editor for IEEE ACCESS and the *Journal of Network and Computer Applications* (Elsevier), an Associate Technical Editor of the *IEEE Communications Magazine*, and an Editor of the *Journal of Cluster Computing* (Springer), *The Computer Journal* (Oxford), the IEEE SDN Newsletter, the *KSII Transactions on Internet and Information Systems*, *Human-centric Computing and Information Sciences* (SpringerOpen), *SpringerPlus*, and *Ad hoc & Sensor Wireless Networks* journal. For more updated information, visit his website at www.attaurrehman.com

**OGUZ BAYAT** received the B.S. degree from Istanbul Technical University, Istanbul, Turkey, in 2000, the M.S. degree from the University of Hartford, CT, USA, in 2002, and the Ph.D. degree from Northeastern University, Boston, MA, USA, in 2006, all in electrical engineering. Before attending the Communication and Digital Signal Processing Research Center at Northeastern University in 2002, he served as an Adjunct Faculty Member with the Department of Electrical and Computer Engineering, University of Hartford. Since 2011, he has been a Professor with the Department of Electrical and Electronics Engineering, Altinbas University. He is currently working as a Technical Leader/Manager on the Nortel project at Airvana Inc., Boston. He has been involved in the research and development of CDMA 1xEV-DO Rev 0, Rev A, and Rev B radio node and radio node controller software products, since 2005. His current research interests include wireless communications, digital signal processing, channel coding, channel estimation, and equalization techniques, in particular multilevel coding, turbo coding, and equalization for wireless communication systems.