IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 October 2022, accepted 4 November 2022, date of publication 14 November 2022, date of current version 21 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221942

RESEARCH ARTICLE

Fast-Yolo-Rec: Incorporating Yolo-Base Detection and Recurrent-Base Prediction Networks for Fast Vehicle Detection in Consecutive Images

NAFISEH ZAREI[1], PAYMAN MOALLEM[1], AND MOHAMMADREZA SHAMS[2]
[1]Department of Electrical Engineering, Faculty of Engineering, University of Isfahan, Isfahan 81746-73441, Iran
[2]Department of Computer Engineering, Shahreza Campus, University of Isfahan, Isfahan 81746-73441, Iran

Corresponding author: Payman Moallem (p_moallem@eng.ui.ac.ir)

ABSTRACT Despite significant advances and innovations in deep network-based vehicle detection methods, finding a balance between detector accuracy and speed remains a significant challenge. This study aims to present an algorithm that can manage the speed and accuracy of the detector in real-time vehicle detection while increasing detector speed with accuracy comparable to high-speed detectors. To this end, the Fast-Yolo-Rec algorithm is proposed. The proposed method includes a new Yolo-based detection network and LSTM-based position prediction networks. The proposed semantic attention mechanism in the spatial semantic attention module (SSAM) significantly impacts accuracy and speed on par with the most recent fast detectors. Recurrent position prediction networks, on the other hand, improve the detection speed by estimating the current vehicle position using vehicle position history. The vehicle trajectories are classified, and the LSTM network for the specified trajectory predicts the vehicle positions. The Fast-Yolo-Rec algorithm not only determines the position of the vehicle faster than high-speed detectors but also allows for the speed control of the detection network with acceptable accuracy. The evaluation results on a large Highway dataset show that the proposed scheme outperforms the baseline methods.

INDEX TERMS Yolo-based detection network, attention mechanism, recurrent prediction network.

I. INTRODUCTION

The detection and classification of vehicles in intelligent transportation systems play a crucial role in urban traffic management, reducing traffic violations, measuring vehicle speeds, and enabling more detailed violation evaluation [1], [2], [3], [4].

Convolutional neural networks (CNNs) are among the most successful methods for promoting object detection. They are excellent at learning image features and can perform various tasks related to classification and bounding box regression [5]. CNN-based methods are divided into one-stage and two-stage detection networks. One-stage detectors have high inference speeds. On the other hand, two-stage detectors have significant localization accuracy and low speed. These networks have revolutionized the detection of objects; however, despite recent advances, the tradeoff between the speed and accuracy of such networks remains a major challenge [6]. Two-stage detection networks, such as R-CNN [7], Fast R-CNN [8], and Faster R-CNN [9], focus on areas of the image that are more likely to contain the target rather than the entire image. Each of these areas is processed separately by the network. Due to a large number of selected regions, in the range of 1000 to 2000, the network can process the image thousands of times. While single-stage detectors, such as Yolo [10], process the image only once. Therefore, single-stage networks run much faster.

Several variants of YOLO have been introduced in recent years, including YOLOV2 [11], YOLOV3 [12], YOLOV4, and YOLOV5 [13], [14], [15]. Although the new versions of YOLO are more accurate, their execution speed is not noticeably faster than the basic model. This paper presents a Yolo-based method called Fast-Yolo-Rec to address this issue. In this algorithm, the position of the vehicles is

The associate editor coordinating the review of this manuscript and approving it for publication was Angel F. García-Fernández.

predicted in several input frames, and in some of them, detection is used to determine the position of the vehicles. This is because predicting the situation of vehicles is done much faster than detecting them, which increases the average speed of implementing the Fast-Yolo-Rec algorithm. On the other hand, the accuracy of the presented algorithm is maintained due to the use of the detector network in this algorithm. These networks complement each other and help maintain accuracy and increase speed.

The proposed detector in Fast-Yolo-Rec uses a semantic attention mechanism. This mechanism improves vehicle positioning accuracy and reduces the target miss rates based on its superior performance. The accuracy of the proposed detector in the Fast-Yolo-Rec algorithm is comparable to the last versions of YOLO. Still, its number of parameters is less, and its speed is higher. The reason for achieving this desired result is using the meaningful attention mechanism, which is implemented using U-Net-based segmentation networks. In this mechanism, feature maps are generated so that they make a significant distinction between the vehicles and the image background. Until now, such a mechanism has not been used in YOLO family detectors. This mechanism makes vehicle positioning more accurate, and the miss rate of the target is reduced. Despite this mechanism, there is no need to deepen the network to achieve higher accuracy. Thus, many problems caused by deepening the network, such as overfitting and high hardware volume, are solved. On the other hand, the speed of the detector does not sacrifice its accuracy.

In the Fast-Yolo-Rec algorithm, motion prediction is also used in addition to detection. In motion prediction, only helpful information is processed in sequential images. Consecutive images in the traffic control system have much redundancy due to the stability of surveillance cameras and the existence of a common background in their recorded images. The reason for the high speed of prediction compared to detection is the elimination of these redundancies.

Traditional methods that use motion include optical flow and the use of differential images. [16], [17], [18]. Differential methods have several significant drawbacks. They cannot detect the position of stationary objects and are not suitable for detecting slow and fast objects. Furthermore, when the background of an image changes, they mistakenly assume the change is a moving object.

Optical flow-based detection methods calculate the direction and velocity of each pixel in an image and use them to separate the moving region from the image background [19]. They are highly dependent on the quality of the input images. Considering the mentioned problems, instead of traditional motion-based detection methods, deep neural networks are used for motion prediction, which also finds long-term dependencies and therefore has higher accuracy.

To improve the prediction accuracy, before using deep neural networks, the trajectory of each car is classified, and it is determined whether the car moves in a straight line or changes its direction to the right or left. Then, predictive networks are performed according to the trajectory specified by the classifier. Another solution is the shortening of the prediction time. Research [20], [21], and [22] shows that shorter forecasting time increases forecasting accuracy. Therefore, in this study, prediction is done only in even frames, and detection is used in other frames. Although, according to the complexity of the scene, the time of prediction and detection can be changed and the accuracy and speed of the algorithm can be managed. In fact, the balance between accuracy and speed is one of the challenges of the detection problems that have been addressed in this study. Overall, the main contributions of the present work include:

- It proposes a flexible algorithm (Fast-Yolo-Rec) in terms of speed and accuracy to find vehicle positions. Depending on the complexity of the image and the predefined speed and accuracy, only the proposed detection network or the integration of the proposed detection and prediction networks can be used in an alternating period. In this research, these two networks are used alternatingly. The detection network is used in primary and odd frames, and the prediction network is used in even frames. Since the predicted network is faster than the detection network, the algorithm speed increased. The prediction network accuracy are improved by regularly using information from the detector in specified frames. These two networks complement each other and improve speed and accuracy.
- A Yolo-based detection network (SSAM-YOLO) that has the following advantages over high-speed one-stage detection networks:
    - It decreases the hardware requirement and accelerates detection by reducing the number of learnable parameter.
    - It improves the detection accuracy using a novel semantic attention mechanism (unlike in popular Yolo-based detection networks) and more effective feature maps where vehicles are effectively differentiated from the background.
    - The scale change robustness of the detection network is increased by using detection heads with two different scales, 13 × 13 and 26 × 26, and the multi-receptive field block (MRF block) in the backbone of the detector. Transferring feature maps of different receptive fields created by parallel paths in the MRF block facilitates the detection of objects of different sizes.
- It provides an efficient and effective algorithm for vehicle trajectory classification and vehicle position prediction based on LSTM recurrent networks and regression. It is more accurate than traditional trajectory predicting methods.

The SSAM-YOLO detection network has comparable accuracy to the last version of YOLO detectors. The baseline Highway category from the CDNet2014 dataset is used to train and evaluate the proposed detector. The proposed detection network has almost the same level of accuracy as YOLOV4_Tiny but 29.38% fewer parameters. It also has 31% fewer parameters and 46% fewer floating point operations than YOLOV7_TINY. Therefore, it has higher speed at the same accuracy. The feature maps produced by the proposed attention mechanism, which successfully separates vehicles from the background, enable the efficient reduction of this parameter. Despite this module, the network is not deepened to obtain better features. The use of fewer parameters accelerates the algorithm and downsizes the hardware. Using the predictor and the SSAM-YOLO detector allows the proposed Fast-Yolo-Rec to take advantage of both methods simultaneously and achieve high accuracy in vehicle detection in addition to the appropriate speed. The results obtained from the implementation of the proposed method and its comparison with the baseline methods are discussed in the evaluation section.

II. RELATED WORK

In recent decades, vehicle detection has greatly interested machine vision researchers. Cameras have increasingly advanced in terms of hardware. They are today more cost-effective and widely used in traffic control systems.

There are two broad categories of vehicle detection methods: traditional and deep network-based. Models based on colors, object contours, and optical flow [23], [24] and background modeling algorithms, such as a mixture of Gaussian (MOG) and its subtraction from the input image [25], [26], HOG and Haar-like features [27], Generalized-Huff, and Kalman filter are commonly used as methods of the former group to detect and predict the positions of vehicles [28], [29]. Principal component analysis (PCA) is employed to provide more efficient data, and support vector machines (SVMs) are used to classify data [30], [31].

In the latter group of methods, features are determined automatically with the help of deep networks. The performance of such features depends on the training dataset, the type, and the effectiveness of the network. A larger number of training data and higher relevance lead to higher accuracy. Networks-based detection methods are divided into two-stage and one-stage groups. Two-stage methods, such as RCNN, are more accurate; however, they lack sufficient speed in real-time applications. They use region search [32] in the image and convolutional network, have a long training time, and require large memory. Mask RCNN, FPN, and R-FCN [33] have improved the feature extraction and classification efficiency of convolutional networks.

Single-stage networks, such as SSD [34] and YOLO, are faster and less accurate than two-stage methods. SSD utilizes MutiBox [35], Region Proposal Network (RPN), and multi-scale representation methods to more accurately locate an object. The YOLO network divides an image into a set of grids. Each grid is responsible for predicting objects whose center points are located within the grid. YOLO variants, e.g., YOLOV2, YOLOV3, and YOLOV4, have been introduced. YOLOV2 improves the YOLOV1 using Darknet-19 as the backbone and anchor boxes to predict the bounding boxes and batch normalization, which normalizes the input of each layer and accelerates network convergence. YOLOV2_Tiny is a very efficient and effective variant in real-time applications. YOLOV3 detects objects at different scales. Thus, it is slower than YOLOV2 and has higher scale change robustness. Moreover, YOLOV4 improves YOLOV3 using CSPDarknet53. Then the concept of a decoupled head was introduced in YOLOX. It has been updated to use a decoupled head and achieve higher accuracy. There are variations of YOLOX split into two categories; Standard Models for high precision and Light Models for edge devices. YOLOX-s is able to achieve the same accuracy as YOLOv4 with half the processing time [36].

YOLO v5 uses Cross-Stage Partial Connections with Darknet-53 as the Backbone and Path Aggregation Network as the Neck, just like the YOLO v4. The significant improvements include novel mosaic data augmentation and auto-learning bounding box anchors. Based on YOLOv4, [14] proposes a YOLOV4-5D network for improving detection accuracy. The backbone network in YOLOV4-5D is the SPDarknet53_dcn(P). The last output layer in the CSPDarknet53 is replaced with deformable convolution to enhance the detection accuracy. In YOLOV4-5D, a new feature fusion module (PAN++) is designed, and five scale detection layers are used to improve the detection accuracy of small objects.

According to the benchmarking performed by Meituan's team, YOLOv6 outperforms YOLOv5 in terms of accuracy and speed. YOLOv6 uses the EfficientRep backbone. Unlike the previous YOLO architectures, which use anchor-based methods for object detection, YOLOv6 opts for the anchor-free approach. This makes YOLOv6 faster compared to most anchor-based object detectors [37]. After that, YOLOV7 was presented by Chien-Yao Wang and colleagues [38]. YOLOv7 enhances object detection by creating a network architecture that predicts bounding boxes more accurately than its competitors at comparable inference speeds. This method uses the extended efficient layer aggregation networks (E-ELAN) to achieve these results. The E-ELAN does not change the gradient transmission path of the original architecture but uses group convolution to increase the cardinality of the added features.

Several works sought to increase the accuracy of detection networks. Tinier-YOLO was developed based on Tiny-YOLO-V3 [39]. The fire module in SqueezeNet is chosen in Tinier-YOLO by looking into the number of fire modules and their locations within the model. A convolutional neural network (CMNet) was proposed for fast vehicle detection in complex scenes [40]. First, it suggests a connect-and-merge residual network (CMRN). Then, a multi-scale prediction network (MSPN) is used to accurately regress the vehicle shape and categorize different types of vehicles.

An intelligent traffic-monitoring system was developed using YOLO and a convolutional fuzzy neural network (CFNN) [41]. It logs the traffic flow on the road. It uses a vehicle-counting technique along with the detection of vehi-

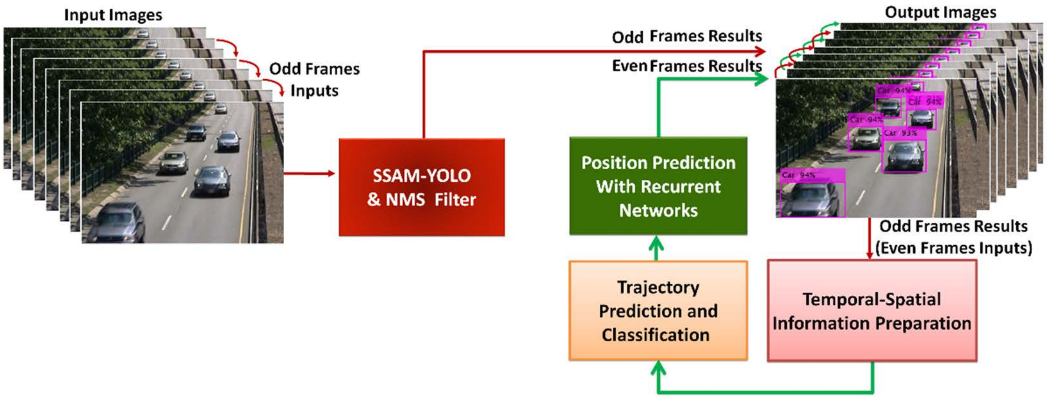

Input Images
Odd Frames Inputs
SSAM-YOLO & NMS Filter
Odd Frames Results
Even Frames Results
Output Images
Position Prediction With Recurrent Networks
Odd Frames Results (Even Frames Inputs)
Trajectory Prediction and Classification
Temporal-Spatial Information Preparation

FIGURE 1. Schematic of the proposed Fast-Yolo-Rec algorithm.

cles to estimate the traffic flow. Then, two efficient models (i.e., CFNN and VectorCFNN) and a network mapping fusion method are proposed to classify vehicles.

A context-exploited method was introduced to integrate features from various receptive fields to obtain contextual representation and improve detection accuracy [42]. To encode the context, it employed multi-branch diverse receptive field module principles.

[43] proposes a Compressed Sensing Output Encoding (CSOE) for detecting pixel coordinates of small objects and crowd counting and localization. The proposed detection framework in [43] consists of a crowd location encoding scheme based on compressed sensing and an end-to-end trainable network made up of observation layers and sparse reconstruction layers and achieves excellent results in scenes with high crowd density. [44] creates an oriented surgical tool with CSLE, a new backpropagation rule for sparse reconstruction, and an end-to-end trainable network. Its approach is quite effective in casting-oriented object localization as regression in encoding signal space.

[45] proposes a dual-branch center face detector (DBCFace). This paper improves face detection via a dual-branch fully convolutional framework without extra anchor design and NMS. It uses two parallel detectors, does not rely on NMS, and achieves similar performance as anchor-based methods with multi-branch. [46] proposes a novel architecture named Serial and Parallel Group Network (SPGNet), which can capture discriminative multi-scale information while keeping the structure compact. Various computer vision tasks, such as image classification, object detection, and person re-identification, have been used to evaluate the SPGNet.

[47] proposes a co-attention scheme containing class-agnostic attention (CA) and semantic attention (SA). These capture object boundary details and global context-aware information from low-level and high-level features. This model can filter out the distracting distraction of background information by fusing these two attentions.

Despite significant breakthroughs in deep learning networks for object detection, the trade-off between detection accuracy and speed remains a challenge. The present study aims to improve the detection speed of the algorithm while maintaining an accuracy comparable to the most recent variants of high-speed detectors, like recent variants of YOLO. Our research also has the ability to locate the occluded vehicle using the pre-blocking areas, which is a benefit. The detector network cannot determine its position when a vehicle is blocked behind an obstacle, such as a larger vehicle or bus. Nonetheless, the predictive network in the proposed Fast-Yolo-Rec can quickly locate the location. Thus, it can be said that another advantage of our research is that it has the ability to locate the occluded vehicle based on the pre-blocking areas. This study handles this challenge by integrating deep learning networks, such as Yolo-based convolutional networks, classifiers, recurrent networks, and segmentation networks.

III. PROPOSED ALGORITHM

Figure 1 depicts the proposed algorithm. It uses two distinct deep networks to find vehicle positions. One deep network is implemented in odd frames (SSAM-YOLO), while the other is executed in even frames. These networks include the recurrent vehicle position prediction network and a YOLO-based detection network. The detection network uses the semantic attention mechanism and is executed in the first 64 frames of consecutive images and odd-numbered frames. As prediction is faster than detection, using a prediction block accelerates the algorithm. In even frames, time-series data should be provided for the trajectory prediction network before using the prediction network. These accurate data are obtained from the detection network in odd frames and increase the accuracy of the prediction network. A rise in the prediction time usually causes errors. The results in this study depict that for a prediction time below 28 frames, the position prediction error is below two pixels, which is desirable. Frames are used alternatingly in the prediction network, and the prediction time of one frame has a very good accuracy.

Therefore, the proposed algorithm has a high speed without losing accuracy, which is not the case with even the fastest detectors. In addition to the time of prediction shortening to

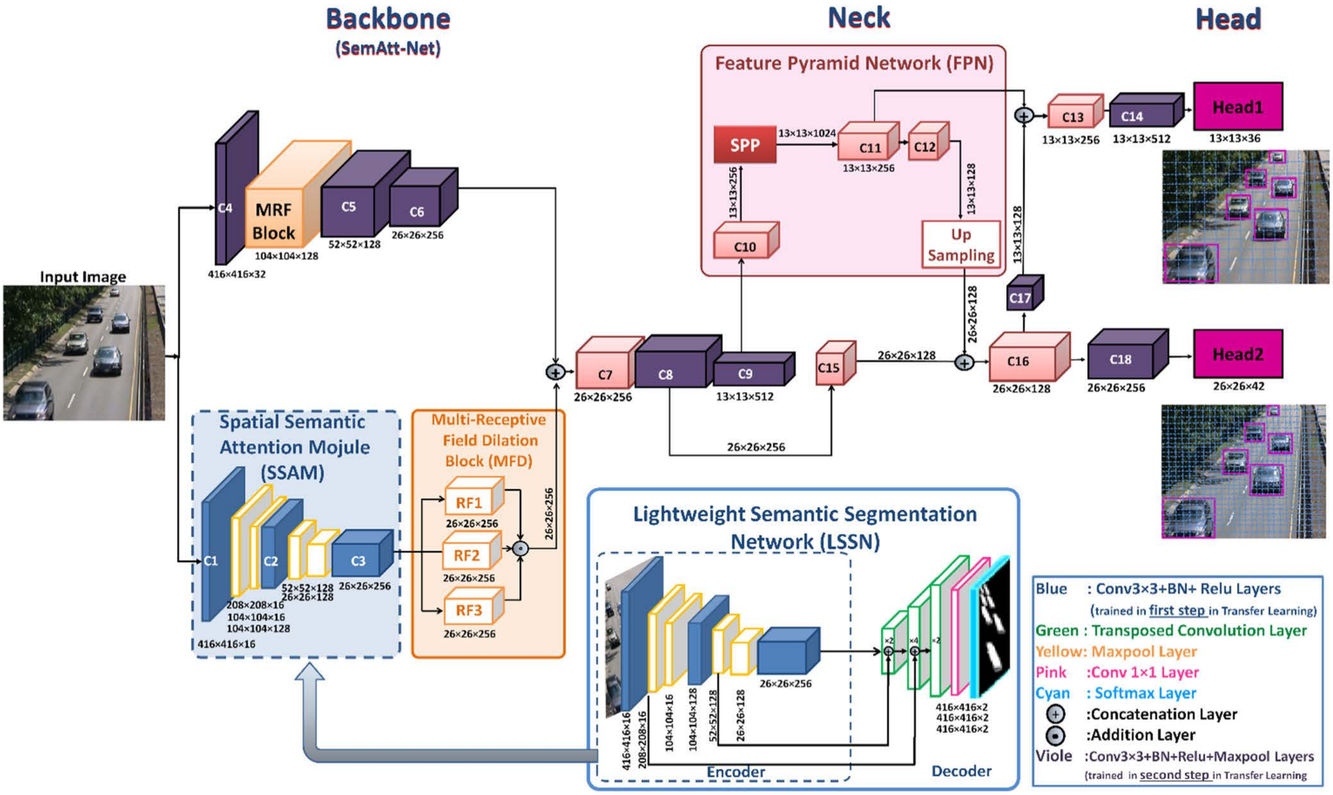


FIGURE 2. Schematic of the proposed SSAM-YOLO detector in Fast-Yolo-Rec algorithm.

increase accuracy, the vehicle trajectory is identified using a classification network. Then, according to the recognized trajectory, the position of the vehicle is determined through the prediction network trained for that trajectory. The classification network identifies whether the vehicle moves straight or takes a lane change. Then, based on the direction identified by the classifier, the recurrent network trained for the corresponding direction is used.

A. DETECTION NETWORK

SSAM-YOLO is proposed as a Yolo-based detection network, as shown in Figure 2. The backbone, neck, and head are the three main components of YOLO-based detectors.

Due to their higher resolution and more accurate spatial features, the extracted feature maps of the backbone are more effective for vehicle detection than for other feature maps in the detection network. The head and neck are more helpful in classifying vehicles since they provide higher semantic data and depth, despite lower spatial detail due to lower resolution. This paper proposes a design for the SSAM-YOLO detector that improves the accuracy of vehicle position detection and increases scale change robustness in light of the MRF block and SSAM module in the backbone, referred to as the Semantic Attention Network (SemAtt-Net).

The MRF Block is constructed to improve feature map extraction at a 104*104 resolution with various receptive fields. In this block, residual connections are used. A structural comparison based on the residual connections between multiple blocks is shown in Figure 3. Figure 3(a) indicates the residual-based block in the YOLOV3 detector, in which two series residual branches are used. Figure 3(b) illustrates the utilization of two residual branches in the CSP blocks of the YOLOV4 detector. Figure 3(c) depicts the structure of the proposed MRF block. The backbone of YOLOV4_Tiny consists of three CSP blocks. The input and output of this block are concatenated by two connections, one of which is the residual connection, and the other one is composed of a slice layer, two 3 × 3-layer convolutions, and a layer with a factor of 1 × 1 convolutions. In the second connection, the layers are successive, and the output attribute maps of the CSP block in this connection are calculated in the 5 × 5 received field.

The proposed MRF block includes one residual connections, two 3 × 3 convolution blocks, one atrous convolution block, and one 1 × 1 convolution block. The extracted feature maps in the MRF block are concatenated into a 3 × 3 and a 5 × 5 receptive field and transferred to the following convolutional layers. Feature maps with a 3 × 3 receptive field and those with a 5 × 5 receptive field in the previous blocks are transferred to the next layer through the MRF block used instead of the earlier blocks.

Thus, feature maps include more spatial details due to their smaller receptive field. The atrous layer also considers

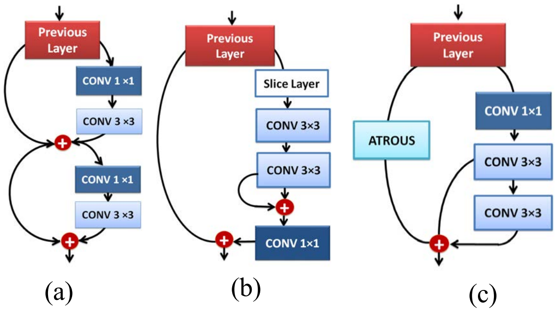


FIGURE 3. Comparison of blocks with residual branches between (a) two residual blocks in the sequence in YOLOV3, (b) a CSP block in YOLOV4 with two residual blocks, and (c) proposed MRF block.

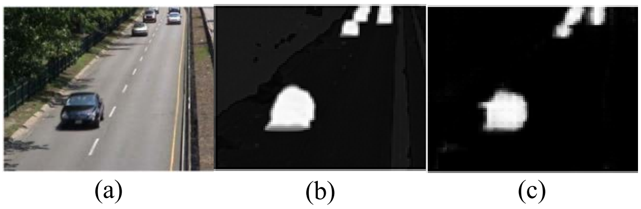


FIGURE 4. Comparison of segmentation networks for (a) the input image, (b) complete segmentation network, and (c) proposed LSSN network.

another 5 × 5 receptive field with different detail. Therefore, the proposed detector effectively detects vehicles of different scales, a challenge in conventional detection networks.

Apart from the MRF block, the SSAM module is used to improve feature maps in the backbone of the SSAM_YOLO detector. The SSAM module in SSAM_YOLO is a UNET-based semantic segmentation network encoder known as LSSN, which was very lightly designed in terms of the number of parameters. This network helps the detector generate desired feature maps in its backbone. Vehicles in these maps are well distinguished from the background, increasing the precision of the detector.

LSSN receives independent training (first training stage). The encoder is then employed as the SSAM module in the detection network. The SSAM module in the detection network pays attention to the spatial positions of vehicles in feature maps with a resolution of 26 × 26 and effectively distinguishes between (foreground) and background vehicles. The detection and LSSN segmentation networks are trained using the same images. As a result, SSAM_YOLO can be trained using transfer training. Also, when training SSAM_YOLO, the learning rate in the SSAM module is zeroed, and the remaining detector training is carried out (second training stage).

As mentioned, the LSSN network is designed and trained to use its encoder in the SSAM_YOLO detector; however, feedback from the encoder to the decoder in LSSN decelerates the detector. Hence, such feedback is eliminated in the SSAM module. To reduce the feedback dependence of trainable parameters in LSSN, minimal feedback is used. Moreover, to decrease the number of SSAM module parameters and accelerate the detector, convolution layers at resolutions 52 × 52, 104 × 104, and 208 × 208 were eliminated from the complete UNET network, utilizing only the pooling layer to obtain a lower resolution.

Figure 4 compares the proposed light semantic segmentation network (LSSN) to a complete segmentation network (CSN) with more feedback and convolution layers in output. As can be seen, the reduction of feedback and convolution layers decreases segmentation accuracy on foreground boundaries in the output of the LSSN network; however, boundary accuracy in the final layer of LSSN with a resolution of 416 × 416 has no significant effect on the output of the SSAM module with a resolution of 26 × 26 (due to high pooling); But the number of parameters and network feedback quantity in LSSN is dropped dramatically, and the use of its encoder in the backbone of the detector accelerates detection.

In the output feature maps, only the central pixels of the cars are separated from the background. Therefore, the MFD block is designed to pay more attention to the cars in the feature maps. The MFD block consists of three maximum pooling layers with receptive fields 2×2, 3×3, and 4×4 (RF1, RF2, and RF3 in Figure 2). This block uses multiple receptive fields for maximum pooling with stride = 1 (no pooling) and has a dilation-like function to ensure that full attention is paid to pixels corresponding to vehicles. More attention improves vehicle detection accuracy and reduces target miss rates.

Figure 5(a) depicts a 416 × 416 input image of the SSAM_YOLO network, while Figs. 5(b)-5(d) show the output feature maps of the convolution layer C6, SSAM module, and MFD block shown in Figure 2. The foreground (vehicles) is effectively differentiated from the background in Figs. 5(c)-5(d).

As can be seen, the proposed SSAM module, MFD block, and transfer learning used to train the SSAM_YOLO detection network provide more effective features than standard Yolo detector convolution layers (figure 5(b)). Finally, concatenating the feature maps from figure 5(d) to the feature maps from figure 5(b) and applying them to the next layer improves detection accuracy.

In this study, other factors improve vehicle detection accuracy, including the selection of suitable anchors. YOLO can function efficiently when several objects are associated with one grid cell. However, in the case of an overlap, where one grid cell contains two different objects, anchor boxes can be used to enable one grid cell to detect several objects. To increase detection accuracy in the proposed SSAM_YOLO detector, the number and size of anchors were chosen more accurately from a different perspective than in other detectors.

Typically, the number of anchor boxes is determined by the number of object classes detected. In this study, 11 anchors were chosen based on the mean intersection over union to have greater scale robustness in detection.

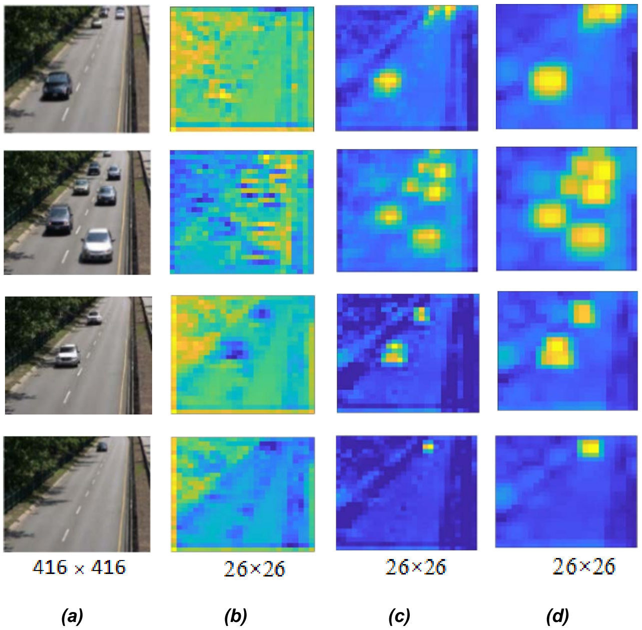


FIGURE 5. Comparison of feature maps for (a) input image, (b) output of convolution layer C6 (Figure 2), (c) output of the proposed SSAM module, and (d) output of the proposed MFD block.

B. TEMPORAL-SPATIAL INFORMATION PREPARATION

The position prediction block uses the history of vehicle movement as time series data and predicts the position of the vehicle in even frames according to it.

$$X = [X^{(t-t_h)}, \ldots, X^{(t-1)}, X^t] \tag{1}$$

where t_h is a fixed (history) time horizon,

$$X^t = [x_0^{(t)}, y_0^{(t)}, x_1^{(t)}, y_1^{(t)}, \ldots, x_n^{(t)}, y_n^{(t)}] \tag{2}$$

where x and y are the vehicle coordinates at time t. Since there are several vehicles in each image, a specific name or unique label should be assigned to each vehicle as long as the vehicle is in the surveillance camera field of view so that the data of each vehicle are stored in a dedicated sub-tensor for the exact vehicle. Let $[x_0^{(t-t_h)}, y_0^{(t-t_h)}, \ldots, x_0^{(t)}, y_0^{(t)}]$ be the position history of the first vehicle and $[x_{\mathrm{n}}^{(t-t_h)}, y_{\mathrm{n}}^{(t-t_h)}, \ldots, x_{\mathrm{n}}^{(t)}, y_{\mathrm{n}}^{(t)}]$ be the position history of vehicle n. The area history of vehicles is $[A_0^{(t-t_h)}, \ldots, A_0^{(t)}]$ to $[A_{\mathrm{n}}^{(t-t_h)}, \ldots, A_{\mathrm{n}}^{(t)}]$, and the intensity average history is $[I_0^{(t-t_h)}, \ldots, I_0^{(t)}]$ to $\left[I_{\mathrm{n}}^{(t-t_h)}, \ldots, I_{\mathrm{n}}^{(t)}\right]$. The similarity distance criterion (SDC) for a vehicle with features $[x^{(t+1)}, y^{(t+1)}, I^{(t+1)}, A^{(t+1)}]$ can be defined as:

$$DPOS0 = \left(x^{(t+1)} - x_0^{(t)}\right)^2 + \left(y^{(t+1)} - y_0^{(t)}\right)^2 \tag{3}$$

$$DAR0 = \left(A^{(t+1)} - A_0^{(t)}\right)^2 \tag{4}$$

$$DIN0 = \left(I^{(t+1)} - I_0^{(t)}\right)^2 \tag{5}$$

$$D^0 = \sqrt{\alpha\,(DPOS0) + \beta(DAR0) + \gamma(DIN0)} \tag{6}$$

$$DPOSN = \left(x^{(t+1)} - x_n^{(t)}\right)^2 + \left(y^{(t+1)} - y_n^{(t)}\right)^2 \tag{7}$$

$$DARN = \left(A^{(t+1)} - A_n^{(t)}\right)^2 \tag{8}$$

$$DINN = \left(I^{(t+1)} - I_n^{(t)}\right)^2 \tag{9}$$

$$D^n = \sqrt{\alpha\,(DPOSN) + \beta(DARN) + \gamma(DINN)} \tag{10}$$

$$SDC = [D^0, \ldots, D^n] \tag{11}$$

$$label = \left\{ \begin{array}{l} \arg(SDC)\min(SDC) < Th \\ n+1 \min(SDC) \geq Th \end{array} \right\} \tag{12}$$

where D^0 is the similarity distance with the first vehicle seen in the scene, D^n is the similarity distance with the vehicle n in the background, and SDC contains all of them. A vehicle label is determined by the minimum value of the SDC and its index. As can be seen, the similarity distance between two vehicles is calculated by the square of the distance between the positions, the average intensity, and their area. Here, α, β, and γ are hyper parameters.

Figure 6 illustrates the labelling output of several sequential frames from the Highway dataset. As can be seen, labelling is stable, and the label of each vehicle remained unchanged in consecutive frames. According to these labels, the position of each vehicle in consecutive frames is recorded as the history of the movement of that vehicle specifically for it. Then, it is recorded in the FIFO-like temporal tensor and used in the prediction block.

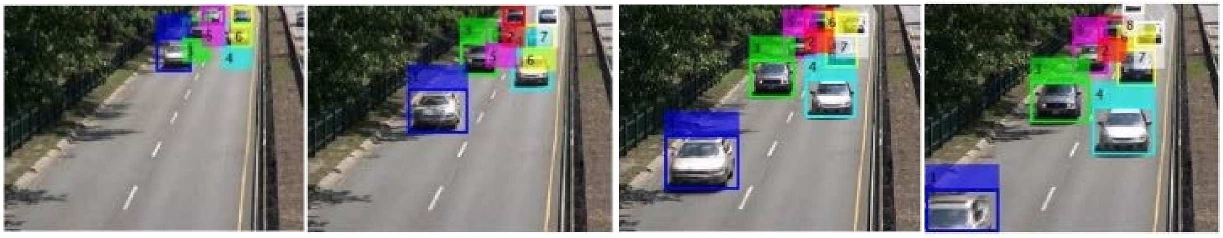
C. POSITION PREDICTION AND TRAJECTORY CLASSIFICATION BLOCKS

In addition to designing a fast and accurate SSAM_YOLO detection network, the position prediction network is presented in this study. The goal of this predictor is to maximize the vehicle position detection speed. Unlike traditional motion prediction methods, LSTM-based recurrent networks account for long-term time dependencies and are thus more accurate than conventional position prediction models, e.g., the constant acceleration (CA) model. The position of a vehicle in the current frame (p_2) is compared to that of the vehicle in the previous frame (p_1) in this model.

$$p_2 = \frac{1}{2}a\Delta t^2 + v\Delta t + p_1 \tag{13}$$

where acceleration a and velocity v are assumed to be known. However, this assumption is not always the case, and the speed and acceleration of vehicles may change many times, depending on traffic flow and driver. In recurrent networks, redirection is learned by the network for different modes of speed and acceleration over time. Since they contain several nonlinear activation functions, they could predict complex movement patterns and trajectories with different accelerations and velocities. These functions determine the data points of the previously saved frames that should be kept or excluded and the data of the current frame that should be added. Assuming h_{t-1} and X_t to be the inputs, the operation of the LSTM network can be formulated as:

$$i_t = \sigma\,(W_i X_t + U_i h_{t-1}) \tag{14}$$



(a) (b) (c) (d)

FIGURE 6. Visual results of the proposed stable labelling on consecutive images with different time interrupt ((a) to (d)).

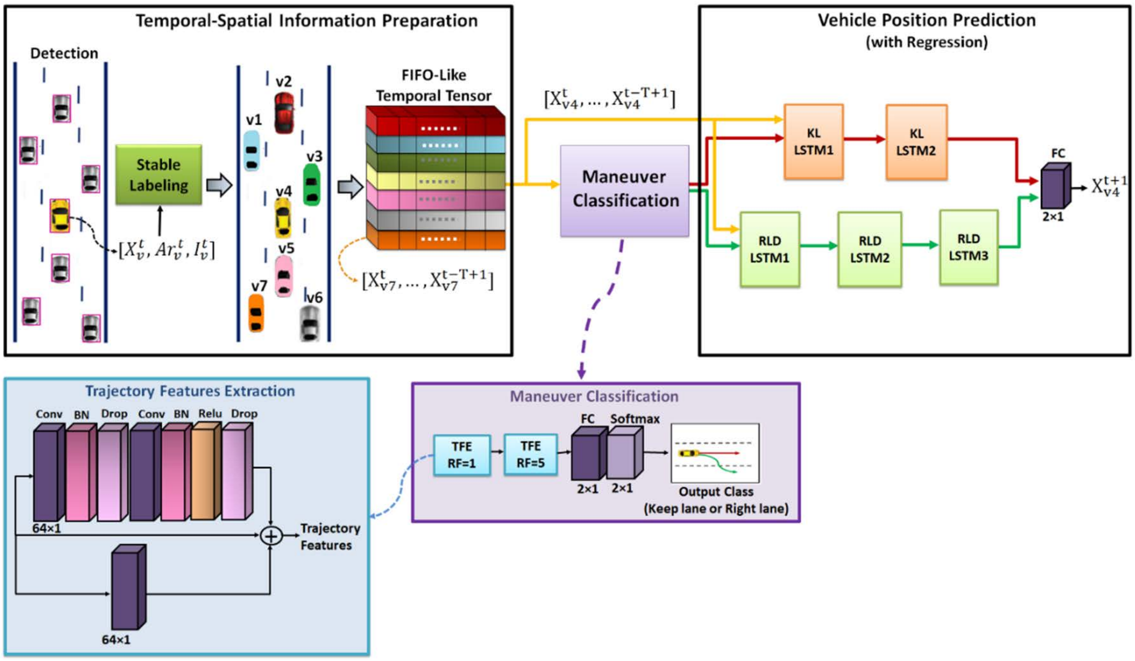


FIGURE 7. Proposed vehicle position prediction architecture.

$$f_t = \sigma(W_f X_t + U_f h_{t-1}) \tag{15}$$

$$o_t = \sigma\,(W_o X_t + U_o h_{t-1}) \tag{16}$$

$$c_t = \tanh\,(W_s X_t + U_s h_{t-1}) \tag{17}$$

$$s_t = i_t \odot c_t + f_t \odot s_{t-1} \tag{18}$$

$$h_t = o_t \odot tanh(s_t) \tag{19}$$

where h_{t-1} is the network output in the preceding frame, while X_t stands for the data introduced in the current frame v.

It contains the historical trajectory $[X_v^{t-T+1}, \dots, X_v^t]$ of the vehicle obtained from the time series data preparation block. Also, "$\odot$" denotes the dot product symbol.

Weight matrices W_i, U_i, W_f, U_f, W_0, and U_0 represent the input, forget, and output gates, respectively, h_t is the network output, and state cell s_t is updated in each frame. The LSTM network is employed to regress vehicle trajectories in two modes of lane keeping and right turns in the Highway dataset. The proposed algorithm architecture is depicted in Figure 7. It includes sections on temporal-spatial information preparation (TSIP), trajectory classification (TC), and vehicle position prediction (VPP). Subsection 3.2 described the TSIP block. The vehicle trajectory is determined in the TC section, which is designed to improve prediction performance.

The classifier determines whether the vehicle is moving in a straight line or changing lanes. The trajectories of vehicles may differ. Given that in this study, there are two categories of straight trajectory and rightward lane change in the dataset used to train recurrent networks, a classifier with two classes is proposed, and KL-LSTM or RLD-LSTM networks are employed to predict vehicle positions based on the classifier-identified trajectory class.

Trajectory classification is carried out by sampling various trajectories learned for each trajectory. It consists of two trajectory feature extraction (TFE) blocks with different receptive fields, a fully connected layer, and a soft-max layer.

The TFE block has three parallel paths. In the first path, the convolution layer is initially used. Let x be the input of the TFE block, and convolution filters are determined using weights w_1 and bias b_1. The first convolution layer output feature map (f_{s1}^1) is written as:

$$f_{s1}^1(x; w_1, b_1) = w_1 *_{s1} x + b_1 \tag{20}$$

where $*_{s1}$ is the convolution operation with stride s1.After the convolution layer, the batch normalization (BN) layer and drop-out layer are exploited. BN plays a key role in avoiding vanishing gradients. For B1 representing a batch of feature maps, BN is defined as:

$$B_1 = batch\left(f_{s1}^1\right) \tag{21}$$

$$\hat{B}_1 = \frac{B_1 - \mu_{B1}}{\sqrt{\sigma_{B1}^2 + \varepsilon}} \tag{22}$$

$$D_1 = Drop\left(\hat{B}_1\right) \tag{23}$$

where $\hat{B}_1, \sigma_{B1}$, and μ_{B1} are the normalized array, variance, and mean of batch B1, respectively. The value of ε is set to 0.001 to prevent null point division. After normalization, the dropout layer is used. The dropout layer is a mask that nullifies the contributions of some neurons to the next layer and leaves others unmodified. Then, another convolution layer is used, the input of which is the dropout output, and convolution filters are determined using weights w_2 and bias b_2. The output feature map f_{s1}^2 is defined as:

$$f_{s1}^2(D1; w_2.b_2) = w_2 *_{s2} D1 + b_2 \tag{24}$$

where $*_{s2}$ is the convolution operation with stride s_2. Then, the batch normalization (BN) layer, ReLU activation layer, and dropout layer are used.

$$B_2 = batch\left(f_{s1}^2\right) \tag{25}$$

$$\hat{B}_2 = \frac{B_2 - \mu_{B2}}{\sqrt{\mu_{B2}^2 + \varepsilon}} \tag{26}$$

$$R = \max\left(0, \hat{B}_2\right) \tag{27}$$

$$D_2 = Drop(R) \tag{28}$$

The use of ReLU helps prevent exponential growth in computation required to operate the classifier network and introduce non-linearity into the BN layer output. The second path in the TFE block is a residual connection used to improve network learning. The third path is a convolutional layer whose receptive field is different in the two TFE blocks; RF=1 in the first block and RF=5 in the next block, while the receptive fields of the first path are the same for the two TFE blocks. The output feature map f_{s1}^3 is defined as:

$$f_{s2}^3(x; w_3, b_3) = w_3 *_{s3} x + b_3 \tag{29}$$

Finally, the feature maps from the first and third paths and the input of the TFE block are combined to form the final feature map out_{MRF}.

$$out_{MRF} = \sum\nolimits_{i=0}^{N} (x\,(i) + D_2\,(i) + f_{s2}^3(i)) \tag{30}$$

The obtained properties are transferred through a fully connected layer *fc_MRF* to the Softmax layer. This layer determines the probability of each of the two trajectory classes.

$$S^{(fc\_MRF_k)} = \frac{\exp(fc\_MRF_k)}{\sum_{k=1}^{N} \exp(fc\_MRF_k)} \tag{31}$$

where N is the length of fc_{MRF} (i.e., 64). After trajectory classification, VPP is executed, where there are two groups of LSTM networks: KL_LSTM and RLD_LSTM. KL_LSTM networks are executed when the straight-line trajectory (Keep-Lane) class is specified in TC, and RLD_LSTM networks are selected when the redirection mode is set to the right. The trajectory is non-linear and more complex in the latter, and RLD_LSTM networks are more than KL_LSTM networks.

IV. RESULTS AND DISCUSSION

This section evaluates the proposed algorithm using Highway data from the CDNet2014 dataset. The algorithm consists of several DNN networks, including the SSAM_YOLO detection network and the lightweight semantic segmentation network LSSN, designed to prepare the SSAM module in SSAM_YOLO. Also, the KL_LSTM trajectory classifier network and RLD_LSTM are used in the algorithm. These networks were trained using the adaptive moment estimation optimization algorithm. The performance of the proposed algorithm is evaluated through comparison to previous works in the average precision (AP), average execution time, and RMSE. The tests were conducted in MATLAB with GPU computing facilities on a PC with a Core i7–3.60GHz CPU, 16GB RAM, GTX1060 GPU, and a Microsoft WINDOWS 10-64bit OS.

A. DATASET

In this study, several deep networks are trained, some of which, i.e., SSAM_YOLO and LSSN networks, are used in odd frames, while the others, i.e., KL_LSTM, RLD_LSTM, and trajectory classification networks, are used in even frames. Due to the design of the SSAM module in the detector, which is responsible for dividing the image into two classes, i.e., vehicles and background, the detection network training requires a dataset suitable for object detection and segmentation tasks.

The KL_LSTM and RLD_LSTM prediction networks are used in even frames, and their input is time-series data that includes the positions of vehicles in consecutive images. Thus, a training dataset of consecutive images is required. And for this reason, the accuracy obtained from our tests is higher and closer to each other than similar references that have used datasets that include scattered and non-sequential images. The Highway dataset from the CDNet2014 dataset contains 1700 consecutive images captured by Highway surveillance cameras, and it also has good ground-truth data for image segmentation. As a result, it is suitable for training the LSTMs and LSSN networks used in this study. However, it does not have relevant data to train the detector. Therefore, the Video Labeler of MATLAB is used to handle the ground-truth challenge of the detector. It enables labelling of the ground-truth data in an image sequence. This app can define rectangular regions of interest (ROI) labels.

B. EVALUATION METHODS

The Highway dataset was used to train the detection network. The performance of the detection network was then assessed using standard detection network evaluation metrics, i.e., AP and average execution time. For various recall levels, AP measures precision. Precision and recall are the two criteria that were used. The precision criterion represents the percentage of true positives. The recall criterion calculates the ratio of true positives to all possible outputs:

$$Precision = \frac{TP}{TP + FP} \tag{32}$$

$$Recall = \frac{TP}{TP + FN} \tag{33}$$

A true positive (TP) is an outcome where the model correctly predicts the positive class of the detected vehicles. Similarly, false positive (FP) and false negative (FN) are the outcomes where the model incorrectly predicts the positive and negative classes, respectively. The performance of the LSTM-based prediction network is evaluated using RMSE:

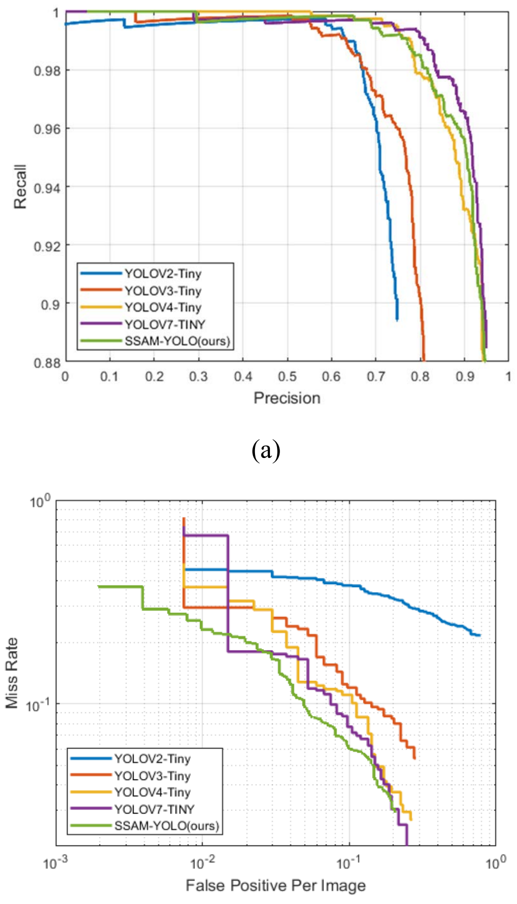
$$RMSE = \sqrt{(\frac{1}{N})\sum_{j=1}^{N}(\hat{x}_j - x_j)^2 + (\hat{y}_j - y_j)^2} \tag{34}$$

where N denotes the number of vehicles in the training set. $[\hat{x}_j^i, \hat{y}_j^i]$ is the predicted position of the vehicle at time step j., and $[x_j^i, y_j^i]$ is the actual position at time step j. These networks were trained for vehicle position prediction on Highway through regression. The average execution time of the Fast-Yolo-Rec algorithm was used to measure the speed of the algorithm in sequential frames.

C. EVALUATION RESULTS

Networks were trained in the Fast-Yolo-Rec algorithm with a batch size of 4 and a total of ten epochs. The learning rate for the first half of the epochs is 0.001 and for the second half of them is 0.0001. This tutorial was additionally optimized using the SGD and Adam algorithms. Figure 8 shows precision, recall, and logarithmic average miss rate to assess the performance of the detection network.

Figure 9 depicts the performance of the detection algorithm for the Highway dataset. The proposed detector considers only the perimeter box with the highest score using the Non-Maximum Suppression (NMS) filter for vehicles with two or more perimeter boxes. The results show that the proposed method is efficient and effective. The MFD block offers improved performance with emphasis on features of the SSAM module, a dilation-like function, transfer learning in SSAM_YOLO detector training, and an optimal number of anchors. Table 1 compares the proposed detector with variants of Yolo-based detection methods on consecutive images in the Highway dataset with one vehicle class. SSAM_YOLO has comparable accuracy to other detectors. As can be seen, its accuracy is almost equal to that of YOLOV4_Tiny and nearly 29.38% fewer parameters than YOLOV4_Tiny.



(a)

(b)

FIGURE 8. Comparison of Yolo detectors and SSAM_YOLO in terms of (a) precision, recall, and (b) miss rate.

The accuracy of YOLOV7_TINY is only 1.18% higher than the proposed SSAM_YOLO detector, while it has a significantly higher computational cost (31% more parameters and 46% more floating point operations). Therefore, the proposed detector has less complexity and more effectiveness. In addition, SSAM_YOLO has a lower target miss rate than other detectors. The output feature maps of the SSAM module show the effectiveness of this module well. On these maps, the vehicle is well distinguished from the background. Distinguishing between target and background is extremely useful for network positioning and lowering miss rates. This implies that the proposed detector has fewer learnable parameters than others, leading to faster training, lower hardware demand, and higher cost-effectiveness. In light of this advantage over YOLOV4_Tiny and being comparable to YOLOV7_TINY, the proposed detector outperforms

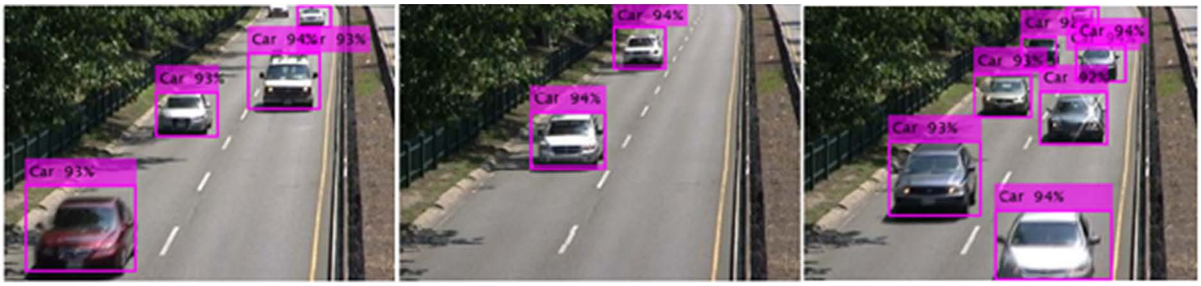


FIGURE 9. Visual results of the SSAM-YOLO detection network on Highway dataset.

TABLE 1. Performance comparison of the proposed YOLO-based detection network and earlier YOLO detection network variants for consecutive images on the Highway (baseline category of CDNet2014).

Yolo-base Detection Methods	Backbone	Input Size	Multi Scale	Attention Mechanism	Average Precision	Parameters	Miss Rate	FLOPs $\times 10^9$
YOLOV2[11]	Darknet19	416	False	False	0.7910	38.73M	0.1567	29.5
YOLOV3[12]	Darknet53	416	True	False	0.8714	49.19M	0.0980	65.8
YOLOV2_TINY [48]	FCCL	416	False	False	0.7420	7.47M	0.344	5.4
YOLOV3_TINY [49]	FCCL	416	True	False	0.8400	7.56M	0.1265	5.5
YOLOV4_TINY [50]	CspDarknet-Tiny	416	True	False	0.9400	6.06M	0.0875	4.3
YOLOX_TINY [36]	CSPDarkNet-Tiny	416	True	False	0.9412	5.06M	0.0740	6.45
YOLOV7-TINY [38]	E-ELAN-based	320	True	False	0.9524	6.2M	0.0700	5.8
SSAM-YOLO (ours)	**SemAtt_Net**	416	**True**	**True**	**0.9406**	**4.28M**	**0.0695**	**3.1**

TABLE 2. Ablation study of the proposed method in terms of average precision (AP) and miss rate.

	AP (%)	Miss-Rate (%)
Detector	88.22	15.14
Detector + SSAM (LSSN)	92.14	9.88
Detector + SSAM +MRF	93.36	7.20
Detector + SSAM +MRF+MFD	94	6.95

YOLOV2 and YOLOV3. The YOLOV2 detector has one head of detection, and the YOLOV3 detector has two heads.

Table 1 also shows that the proposed detector requires 74%, 77%, 39% and 46%fewer floating-point operations (FLOPs) than YOLOV2_TINY, YOLOV3_TINY, YOLOV4_TINY and YOLOV7 TINY, respectively. Table 2 compares the performance of the proposed SSAM-YOLO detector's SSAM module, MRF, and MFD blocks to assess the impact of each detector module on the final output.

The SSAM module improves average precision and miss rate by 3.92% and 5.26%, respectively, while the MRF block improves by 1.22% and 2.68%, and the MFD block improves by 0.64% and 0.25%. Table 2 shows that the combination of three modules in the detector Network achieves an average precision and miss rate of 94% and 6.95%, respectively.

The first five consecutive convolution layers employed by YOLOV2_Tiny and YOLOV3_Tiny are referred to as FCCL Table 1.

TABLE 3. Number of parameters in the first five consecutive convolution layers (FCCL).

Layers	Channel Input	k	Channel Output	Parameters Number
Conv1	3	3	16	448
Conv2	16	3	32	4640
Conv3	32	3	64	18496
Conv4	64	3	128	73856
Conv5	128	3	256	295168
Conv6	256	3	512	1180160

Table 3 reports the number of parameters in various FCCL layers.

In addition to achieving the desired detection accuracy through the SSAM_YOLO detector, vehicles position determination was accelerated through (1) the SSAM_YOLO detector by decreasing the number of parameters and the hardware demand compared to detectors of similar

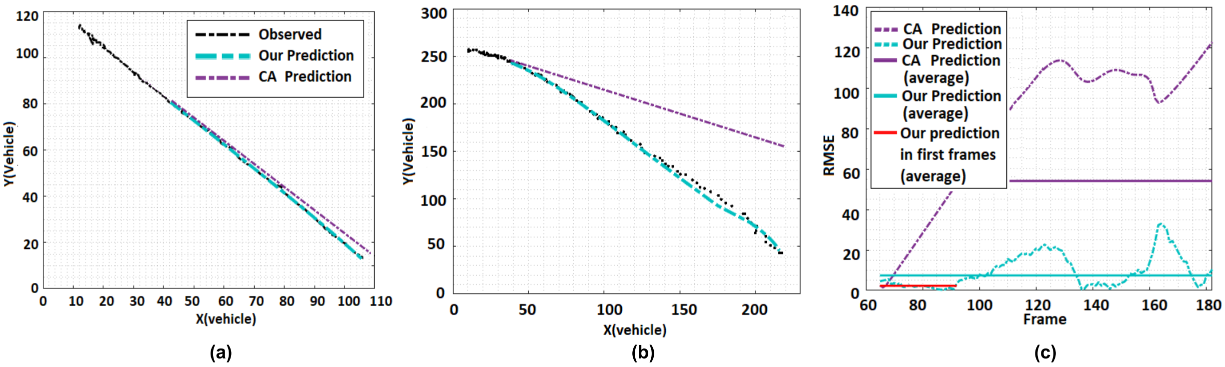


FIGURE 10. Trajectory prediction of two vehicles using the proposed and CA models (the first 64 positions in the trajectories were used for trajectory prediction); (a) the vehicle is driving straight, (b) the vehicle is taking a rightward lane change right, (c) RMSE for the proposed model versus CA model.

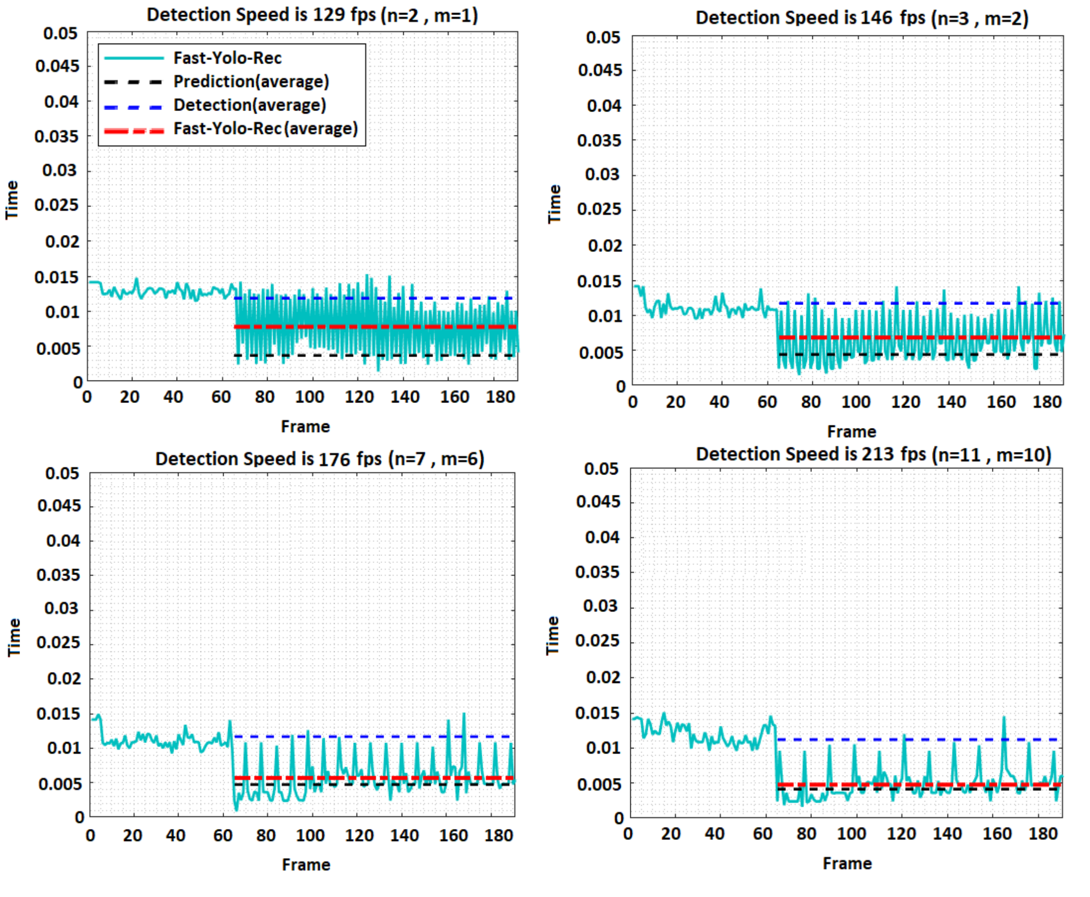


FIGURE 11. Execution time comparison of the proposed Fast-Yolo-Rec algorithm for different n-frame periods at m predictions and n-m detections of SSAM-YOLO.

performance and (2) the periodic prediction using the recurrent networks. The proposed Fast-Yolo-Rec algorithm uses trajectory classification and recurrent networks for vehicle position prediction. As shown in Figure 10, the actual trajectories of two independent vehicles are displayed in black, the predicted position of the proposed model in this study is shown in cyan, and the position prediction of the CA model is indicated in violet. In Figure 10(a), the vehicle moves in a straight line; in Figure 10(b), the vehicle changes lanes to the right. As can be seen, the trajectories are effectively predicted, and the proposed model is more accurate than the CA model. The higher accuracy of the proposed model stems from the LSTM networks and trajectory classification.

The LSTM networks consider long-term time dependencies, in contrast to the CA model. Figure 10(c) depicts the RMSE for a more accurate comparison of the two models.

As can be seen, the prediction error of the proposed model is lower than that of the CA model. The prediction implemented in successive frames without using detector data in alternating frames to compare the two models more rationally. The average RMSE in the proposed Fast-Yolo-Rec algorithm is 50% lower than the RMSE displayed in Figure 10(c) due to using the results of proposed detector with excellent accuracy in odd frames. The results of the proposed algorithm for an *n-frame* period are shown in Figure 11, while prediction was performed *m* times and detection was performed *n-m* times in an *n-frame* period. As can be seen, the prediction and detection networks shortened the average time of the proposed algorithm with a negligible RMSE, as shown in Figure 10 (c). Hence, the accuracy of the algorithm does not undergo a significant change, while detection speed increases significantly.

The red line in Figure 10 (c) shows the average prediction RMSE on the first 28 frames of the prediction. As can be seen, the detection accuracy of the first 28 frames is better than the subsequent frames, and the RMSE error is lower for them. Therefore, prediction accuracy is higher for $n \leq 28$.

The algorithm was executed for *n=2* and *m=1* at 129 fps. In this state, the proposed SSAM_YOLO detection network was executed in odd frames and prediction networks was performed in even frames each at this speed. The algorithm was executed at 146, 176, and 213 fps for (*n=3* and *m=2)*, *(n=7* and *m=6)*, and *(n=11* and *m=10*), respectively.

V. CONCLUSION

The proposed Fast-Yolo-Rec technique accelerates vehicle position detection. It consists of two main parts: 1) vehicle detection network. 2) a maneuver classifier and networks for predicting vehicle position that use an alternating cycle with a certain number of frames. The detector network is used in some frames, and the classifier and predictor networks are used in other frames of this cycle. The accuracy of the Fast-Yolo-Rec algorithm has improved as the increasing the accuracy of the SSAM_YOLO detector, and it has accelerated in light of the vehicle position prediction network. As prediction is faster than detection, the use of detection in the first 64 frames and odd frames and the use of the predictor in even frames of the input images enhances the average speed of the algorithm. Moreover, by using a vehicle trajectory classifier, the accuracy of the prediction network is boosted compared to traditional vehicle position prediction approaches and LSTM networks that are particularly trained based on the output of the classifier.

The accuracy of the SSAM_YOLO detector in the Fast-Yolo-Rec algorithm is also increased through the MRF block, attention mechanism in the LSSN network and SSAM module, MFD block, transfer learning in the training of the SSAM_YOLO network, and the optimal number of anchors. An image contains several vehicles, and the data for each vehicle should be stored in a dedicated array for the same vehicle. Thus, stable labelling is required to obtain the data for the position prediction network. This data includes the vehicle positions over time. This is fulfilled by using the detection network in odd frames, and no extra processing is required. The proposed detection and prediction networks are complementary and assist in finding the positions of vehicles faster than well-known high-speed detectors, leading to sufficient accuracy, speed, and greater flexibility. The flexibility of the proposed Fast-Yolo-Rec algorithm arose from the changing frequency of using the aforementioned networks in a periodic cycle. The real-time vehicle detection performance of the proposed algorithm is demonstrated using a real-life Highway dataset.

REFERENCES

[1] Z. Li and D. Hensher, "Understanding risky choice behaviour with travel time variability: A review of recent empirical contributions of alternative behavioural theories," *Transp. Lett.*, vol. 12, no. 8, pp. 580–590, Sep. 2020.

[2] X. Chen, Y. Chen, and G. Zhang, "A computer vision algorithm for locating and recognizing traffic signal control light status and countdown time," *J. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 533–546, Sep. 2021.

[3] D. Ding, J. Tong, and L. Kong, "A deep learning approach for quality enhancement of surveillance video," *J. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 304–314, May 2020.

[4] N. Mahmoud, M. Abdel-Aty, Q. Cai, and J. Yuan, "Estimating cycle-level real-time traffic movements at signalized intersections," *J. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 400–419, Jul. 2022.

[5] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[6] C. G. Manuel, T.-M. Jesus, L.-B. Pedro, and G.-G. Jorge, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sens.*, vol. 13, no. 1, p. 89, 2020.

[7] 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.*, Jun. 2014, pp. 580–587.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[9] 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. 2015.

[10] 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.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognition*, Jul. 2017, pp. 7263–7271.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[14] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOv4–5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[15] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, Jan. 2022.

[16] P. Singh, B. B. V. L. Deepak, T. Sethi, and M. D. P. Murthy, "Real-time object detection and tracking using color feature and motion," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2015, pp. 1236–1241.

[17] L. Xiao and T.-Q. Li, "Research on moving object detection and tracking," in *Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discovery*, Aug. 2010, pp. 2324–2327.

[18] H. Wang, P. Wang, and X. Qian, "MPNET: An end-to-end deep neural network for object detection in surveillance video," *IEEE Access*, vol. 6, pp. 30296–30308, 2018.

[19] Y. Liu, Y. Lu, Q. Shi, and J. Ding, "Optical flow based urban road vehicle tracking," in *Proc. 9th Int. Conf. Comput. Intell. Secur.*, Dec. 2013, pp. 391–395.

[20] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1468–1476.

[21] Y. Yoon, T. Kim, H. Lee, and J. Park, ''Road-aware trajectory prediction for autonomous driving on highways,'' *Sensors*, vol. 20, no. 17, p. 4703, 2020.

[22] L. Lin, W. Li, H. Bi, and L. Qin, ''Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms,'' *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 197–208, Mar. 2021.

[23] L. H. Pham, T. T. Duong, H. M. Tran, and S. V.-U. Ha, ''Vision-based approach for urban vehicle detection & classification,'' in *Proc. 3rd World Congr. Inf. Commun. Technol. (WICT)*, Dec. 2013, pp. 305–310.

[24] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, ''Vehicle detection using normalized color and edge map,'' *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 850–864, Mar. 2007.

[25] A. Faro, D. Giordano, and C. Spampinato, ''Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1398–1412, Dec. 2011.

[26] K. Park, D. Lee, and Y. Park, ''Video-based detection of street-parking violation,'' in *Proc. Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*, 2007, pp. 152–156.

[27] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, ''A cascade of boosted generative and discriminative classifiers for vehicle detection,'' *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–12, Dec. 2008.

[28] Indrabayu, R. Y. Bakti, I. S. Areni, and A. A. Prayogi, ''Vehicle detection and tracking using Gaussian mixture model and Kalman filter,'' in *Proc. Int. Conf. Comput. Intell. Cybern.*, 2016, pp. 115–119.

[29] V. Rin and C. Nuthong, ''Front moving vehicle detection and tracking with Kalman filter,'' in *Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Feb. 2019, pp. 304–310.

[30] Z. Chen, N. Pears, M. Freeman, and J. Austin, ''Road vehicle classification using support vector machines,'' in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Nov. 2009, pp. 214–218.

[31] Q. B. Truong and B. R. Lee, ''Vehicle detection algorithm using hypothesis generation and verification,'' in *Proc. Int. Conf. Intell. Comput.*, Berlin, Germany, 2009, pp. 534–543.

[32] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, ''Selective search for object recognition,'' *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[33] J. Dai, Y. Li, K. He, and J. Sun, ''R-FCN: Object detection via region-based fully convolutional networks,'' 2016, *arXiv:1605.06409*.

[34] 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.* Cham, Switzerland: Springer, 2016, pp. 21–37.

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

[36] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ''YOLOX: Exceeding Yolo series in 2021,'' 2021, *arXiv:2107.08430*.

[37] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, ''YOLOv6: A single-stage object detection framework for industrial applications,'' 2022, *arXiv:2209.02976*.

[38] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, ''YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,'' 2022, *arXiv:2207.02696*.

[39] W. Fang, L. Wang, and P. Ren, ''Tinier-YOLO: A real-time object detection method for constrained environments,'' *IEEE Access*, vol. 8, pp. 1935–1944, 2020.

[40] F. Zhang, F. Yang, C. Li, and G. Yuan, ''CMNet: A connect- and-merge convolutional neural network for fast vehicle detection in urban traffic surveillance,'' *IEEE Access*, vol. 7, pp. 72660–72671, 2019.

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

[42] S. Xie, C. Liu, J. Gao, X. Li, J. Luo, B. Fan, J. Chen, H. Pu, and Y. Peng, ''Diverse receptive field network with context aggregation for fast object detection,'' *J. Vis. Commun. Image Represent.*, vol. 70, Jul. 2020, Art. no. 102770.

[43] Y. Xue, Y. Li, S. Liu, X. Zhang, and X. Qian, ''Crowd scene analysis encounters high density and scale variation,'' *IEEE Trans. Image Process.*, vol. 30, pp. 2745–2757, 2021.

[44] Y. Xue, Y. Li, S. Liu, P. Wang, and X. Qian, ''Oriented localization of surgical tools by location encoding,'' *IEEE Trans. Biomed. Eng.*, vol. 69, no. 4, pp. 1469–1480, Apr. 2021.

[45] X. Li, S. Lai, and X. Qian, ''DBCFace: Towards pure convolutional neural network face detection,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1792–1804, Apr. 2021.

[46] X. Wang, S. Lai, Z. Chai, X. Zhang, and X. Qian, ''SPGNet: Serial and parallel group network,'' *IEEE Trans. Multimedia*, vol. 24, pp. 2804–2814, 2021.

[47] Y. Wu, S. Feng, X. Huang, and Z. Wu, ''L4Net: An anchor-free generic object detector with attention mechanism for autonomous driving,'' *IET Comput. Vis.*, vol. 15, no. 1, pp. 36–46, Feb. 2021.

[48] H. R. Alsanad, O. N. Ucan, M. Ilyas, A. U. R. Khan, and O. Bayat, ''Real-time fuel truck detection algorithm based on deep convolutional neural network,'' *IEEE Access*, vol. 8, pp. 118808–118817, 2020.

[49] P. Adarsh, P. Rathi, and M. Kumar, ''YOLO v3-tiny: Object detection and recognition using one stage improved model,'' in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 687–694.

[50] Q. Liu, X. Fan, Z. Xi, Z. Yin, and Z. Yang, ''Object detection based on Yolov4-tiny and improved bidirectional feature pyramid network,'' *J. Phys., Conf.*, vol. 2209, no. 1, Feb. 2022, Art. no. 012023.

NAFISEH ZAREI received the B.Eng. degree in electrical and electronics engineering from the Kashan University of Technology. She is currently pursuing the Ph.D. degree with Isfahan University, Iran. Her research interests include the fields of machine vision, image processing, and deep learning.

PAYMAN MOALLEM was born in Tehran, Iran, in 1970. He received the B.Sc. degree in electronics engineering from the Isfahan University of Technology, Isfahan, Iran, in 1992, and the M.Sc. degree in electronics engineering and the Ph.D. degree in electrical engineering from the Amirkabir University of Technology, Tehran, in 1996 and 2003, respectively. From 1994 to 2002, he conducted research for the Iranian Research Organization Science and Technology on the topics, such as parallel processing, robot stereo vision, and DSP boards development. In 2003, he joined the University of Isfahan, Isfahan, as an Assistant Professor, where he was promoted to an Associate Professor and a Full Professor, in 2010 and 2015, respectively. He has authored more than 300 papers published in peer-reviewed journals and conference proceedings, and five books. His research interests include remote sensing, image processing and analysis, computer vision, neural networks, and pattern recognition.

MOHAMMADREZA SHAMS received the B.S. degree from the Isfahan University of Technology, in 2008, the M.S. degree from the University of Tehran, in 2012, and the Ph.D. degree from the University of Isfahan, Iran, in 2017, all in computer engineering. He is currently an Assistant Professor with the Computer Engineering Department, University of Isfahan (Shahreza Campus). His research interests include data mining, text mining, computer vision, and deep learning.