A review of object detection approaches for traffic surveillance systems

Ayoub El-Alami[1], Younes Nadir[2], Khalifa Mansouri[1]
[1]Modélisation et Simulation des Systèmes Industriels Intelligents (M2S2I), Equipe Systèmes Informatiques Distribués, Ecole Normale Supérieure de l’Enseignement Technique (ENSET), Université Hassan II de Casablanca, Mohammedia, Morocco
[2]Modélisation et Simulation des Systèmes Industriels Intelligents (M2S2I), Équipe Technologies de l'Information et Intelligence Artificielle, Ecole Nationale Supérieure de l’Art et de Design (ENSAD), Université Hassan II de Casablanca, Mohammedia, Morocco

Article Info

Article history:

Received Dec 31, 2023
Revised Jul 7, 2024
Accepted Jul 11, 2024

Keywords:

Convolutional neural networks-based object detection
Computer vision
Object detection
Traffic surveillance systems
Vehicle detection

ABSTRACT

With the decreasing cost of traffic cameras and rapid advancement in computer vision and artificial intelligence, developing robust traffic surveillance systems has become more feasible and practical. These systems can easily outperform traditional human monitoring systems, as they can collect and analyze traffic data coming from multiple cameras efficiently. A good understanding of this data allows the detection easily road anomalies in real time and in an autonomous way. Therefore, an intelligent traffic system typically consists of three components: object detection, object tracking, and behavior analysis components. In this paper, we present a review of some of the well-known object detection techniques used in traffic video surveillance. The review begins with a brief introduction to the history of object detection and the evolution of its techniques. Then we review separately the two main approaches of detection, which are traditional and deep learning approaches of detection. Finally, an experimental analysis has been conducted to evaluate and compare the performance of some of the recent relevant detection methods in terms of speed and precision, in detecting vehicles in a traffic scenario.

Corresponding Author:

Ayoub El-Alami
Modélisation et Simulation des Systèmes Industriels Intelligents (M2S2I), Equipe Systèmes Informatiques Distribués, Ecole Normale Supérieure de l’Enseignement Technique (ENSET), Université Hassan II de Casablanca
Mohammedia, Morocco
Email: ayoubalami6@gmail.com

1. INTRODUCTION

An intelligent traffic video surveillance system (ITVS) is a system designed to detect and track moving objects on roads and evaluate their behaviors based on their real-time visual information. However, developing an efficient ITVS is still a challenging problem due to the complexity of tasks required to process and analyze these gathered data. This complexity is related especially to the variety of environmental conditions associated with the surveillance problem such as the speed of moving vehicles, occlusions, and the variation in weather and luminosity. An ITVS usually performs several tasks that are: i) collecting traffic data from traffic network cameras; ii) extracting information about different objects (identities, speeds, and trajectories); and iii) understanding and evaluating objects’ behaviors. Object detection in surveillance tasks is performed by processing real-time video streams captured by one or multiple surveillance cameras mounted on high poles. These cameras constitute the physical infrastructure of the surveillance network. Object detection is considered the first and most important step of the surveillance system, and it is responsible for translating raw video data into meaningful information. This task allows extracting object categories and their spatial parameters such as positions and dimensions. Afterward, the various spatiotemporal parameters of objects such as identity, direction, movement, speed, and acceleration, are calculated in the object-tracking step. In cases of occlusion or temporary disappearance, a necessary re-identification process is added to keep track of missing objects. Finally, the object behavior analyzer evaluates the spatiotemporal parameters of tracked objects to detect abnormal behaviors such as accidents. In addition to its monitoring roles, intelligent traffic video surveillance can provide useful services by sharing the results of its behavioral analysis with drivers, emergency assistance, and law enforcement agencies. In recent years, object detection has become a popular and powerful tool supported by the artificial intelligent and computer vision community. It has contributed to a wide range of fields such as robotics, security, and self-driving vehicles. Due to its importance in any ITVS, in this work, we are interested in the object detection techniques and methods, especially those related to this field of application.

In the literature, two main detection approaches can be distinguished. The first is the traditional detection approach, which includes generally handcrafted algorithms that allow the localization and classification of objects. Several works adopt this approach to detect vehicles on roads and to address traffic surveillance problems. Motion-object detection has been one of the popular methods used for this problem in the early days and used in studies [1]–[5]. In addition to motion detection, several pipelines have been proposed to ensure the recognition of vehicles such as in studies [6]–[11]. These works combine the localization task with a feature extraction to explore objects' representations using handcrafted descriptors such as scale invariant feature transform (SIFT) [12], oriented FAST and rotated BRIEF (ORB) [13], or histogram of oriented gradient (HOG) [14].

Recently, and thanks to the emergence of deep convolutional neural networks (DCNNs), a significantly powerful approach to detection has been popularized, starting with the proposal of the regional-convolutional neural network (R-CNN) detector. This detector operates in two stages: i) selective search that proposes regions of interest (ROIs); and ii) CNN feature extractor that explores ROIs to classify them. Since then, several improvements have been suggested to optimize the two stages of bounding box localization and classification, which lead to more performant models such as Fast R-CNN, and Faster R-CNN. In contrast to two-stage detection, single-shot detector (SDD) and you only live once (YOLO) have been proposed as one-stage detectors that can accomplish detection tasks through one pass by the network and via a fixed grid. This processing way allows reducing significantly the detection time compared to previous architectures. Regarding the task of vehicle detection, the majority of recent works DCNN architectures as the backbone of their approaches, such in studies [15]–[18]. In this paper, a systematic review is provided to summarize the two main approaches and their different algorithms and models proposed to address the problem of traffic surveillance. In addition to this review, we provide an experimental comparison and analysis of the recent version of YOLO detectors in terms of speed and accuracy. In our experiment, we focused only on comparing recent YOLO versions due to their remarkable performances that largely outperform previous architectures. The results show that models' performance varies considerably according to their sizes, since small models like recent Nano versions of YOLO, perform faster compared to larger ones, but with less accuracy.

One of the main challenges in traffic surveillance systems is the detection of vehicles and pedestrians with high accuracy and real-time performance, especially when operating on resource-constrained hardware. Achieving this balance requires optimizing detection algorithms to function efficiently without compromising on speed or precision. Therefore, this review has been conducted in the context of developing a traffic surveillance system specifically designed for deployment on such devices. In this study, we investigate recent well-known detection approaches and establish a performance comparison of popular detectors to identify the most suitable candidates for resource-constrained traffic surveillance systems. The comparative results reveal that the small and nano models of YOLOv5 and YOLOv8 offer the best performance among the evaluated models. Specifically, the small models exhibit a significant advantage in precision, making them ideal for situations requiring notable accuracy. In contrast, the nano models demonstrate remarkable speed, highlighting their suitability for real-time processing in resource-constrained environments. This dual advantage can be capitalized upon, by proposing for example a hybrid approach that utilizes both models in an adaptive manner to tackle this problem. This approach could leverage the high precision of the small model and the remarkable speed of the nano one, by dynamically switching between them based on the specific requirements of the surveillance task at hand.

Our main contribution can be summarized as follows: discussing object detection approaches and reviewing relevant approaches used for traffic surveillance; conducting experiments to compare the performance of recent YOLO detectors in terms of speed, accuracy, and precision. The rest of this paper is organized as follows. In section 2, traditional and DCNN object detection approaches are presented. In section 3 we present the review of DCNN detection approaches applied for the traffic surveillance task.

Afterward, we present the results of conducted experiments in section 4. At the end, some concluding observations are presented in section 5.

2. OBJECT DETECTION

Object detection can be described as a combination of localization and classification tasks, the first task indicates the spatial information of objects using bounding boxes, whereas the second one aims to recognize the categories of them. An efficient object detection system must be able to detect appearing objects within a reasonable time and with good accuracy. In the literature, two main approaches can be distinguished for object detection and categorized into two classes: traditional methods-based detectors and deep learning-based detectors. Object detectors have evolved over several years to meet the precision and rapidity required by different kinds of demanding applications such as real-time surveillance systems. In the following, we will focus on discussing relevant techniques and methods of traditional and deep learning-based detectors as well as some studies based on them to address the problem of vehicle detection in traffic surveillance contexts. Figure 1 shows our proposed classification of relevant algorithms and approaches used to address the problem of detection. In this classification, we show and classify the most popular methods adopted for the traditional pipeline of detection, that consists mainly of localization, classification, and feature extraction operations, and that are implemented in the majority of studies. In addition to that, the figure also shows the recent and relevant CNN models of the two main end-to-end approaches of deep learning-based detectors.

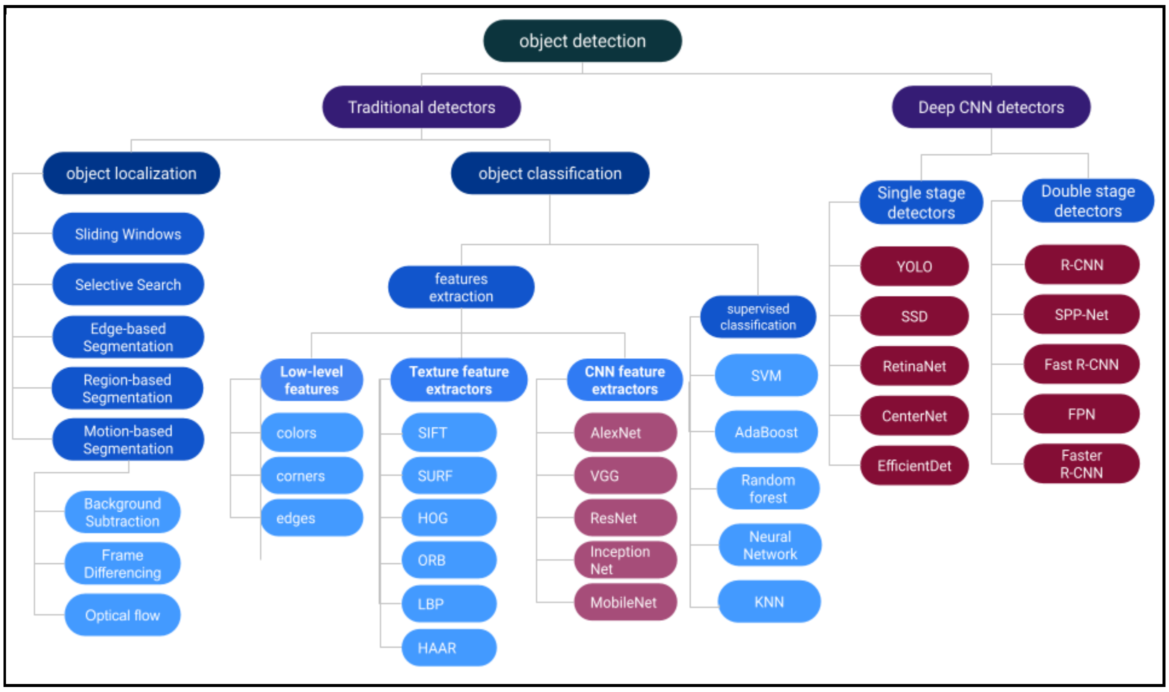

Figure 1. Classification of relevant object detection methods

2.1. Traditional object detection

Traditional object detection approaches refer to handcrafted methods that operate based on a pipeline the three tasks: object localization, feature extraction, and image classification. This sort of detection approach was dominant for a long time before the advent of deep learning and CNN-based techniques. In the following, we will discuss different components of traditional techniques as well as some relevant traditional-based works that address the problem of traffic surveillance monitoring.

2.1.1. Object localization

Object localization, also known as candidate region proposal, is described as a class-agnostic object detection task and refers to the process of determining the locations of objects of interest within an image using bounding boxes. Several approaches have been proposed for this purpose that generally based on:

i) Straightforward exhaustive strategies that explore all possible regions to look for objects such as the sliding windows technique. These methods are considered computationally intensive; alternatively, ii) it can be based on image segmentation strategies, which consist of partitioning an image into multiple segments or blobs by clustering pixels based on appearance features such as colors, textures, or edges to isolate different regions; iii) Another popular method is the selective search, which consists of combining the strengths of sliding windows and image segmentation approaches. Selective search is done through clustering and merging of close similar regions on multiple scales and hierarchical ways based on their similarity. One of the advantages of this last approach is providing rapidly a small number of accurate proposals of multiple scales and orientations [19]; and iv) Another popular method is motion-based segmentation that extracts the motion features of pixels of moving objects. It consists of using a background modeling operation performed over multiple frames to differentiate moving objects effectively. In the following, we will discuss in details motion-based segmentation since it was one of the most effective and popular approaches to detection the era of deep learning.

Motion-based segmentation is one of the widely used localization approaches that are suitable for scenarios where the background is stationary. Its main idea consists of taking advantage of the motion characteristic of foreground moving objects compared to the stationary background in a video scene. The pipeline of this approach consists typically of two steps: The first one is motion detection which consists of computing the optical flow between two or more consecutive frames and that describes the pattern of apparent motion of individual pixels. The second one is motion segmentation which occurs once the motion has been detected and aims at segmenting moving pixels and separating them from the background. This can be achieved by a thresholding operation over the optical flow magnitude, and a clustering algorithm to group pixels that belong to the same object. This motion-based detection allows the generating of a set of candidate regions or blobs in a video scene that can be refined afterward based on their feature, color, texture, or shape to improve the accuracy of the proposals. Some of the relevant motion-based techniques are frame differencing, background subtraction, and optical flow. The frame differencing algorithm is the simplest approach that operates by calculating the difference between two successive frames on pixel-level. Otherwise, background subtraction is a motion segmentation technique that separates foreground objects from the background by performing a subtraction between an image and an estimated background model. For both previous methods, a thresholding operation step allows ignoring small variation in pixel intensities. This approach increases the accuracy by ignoring small variations occurred due to environmental conditions like lighting conditions.

However, despite its effectiveness, motion-based segmentation still has some limitations that affect its accuracy in detecting moving objects in specific situations. Some of these limitations include the sensitivity of the method to the slightest camera motion, small changes in illumination, or unintended background motion, such as the natural motion occurring as a result of tree and leaf movements. Another limitation is the accidental false detections of shadows that might be considered as separate moving objects, especially when adopting an appropriate threshold, or in some special luminosity conditions. However, the most relevant drawback of motion segmentation is the loss of detection when the motion of an object is interrupted or if the object of interest has the same color or texture as the background. Motion-based segmentation also faces difficulties in handling occlusions that occur when an object is partially or completely overlapped by another one. These situations usually yield inaccurate object segmentations and consequently cause misinterpretation of scenes that affect detection reliability. Therefore, when using this technique in traffic surveillance systems these limitations may affect the ability to detect stationary or occluded vehicles or vehicles with the same color or texture as the road. To address these issues, additional techniques can be considered and combined with motion-based segmentation to improve its robustness which is based on appearance-based information.

2.1.2. Object classification

Once the localization is done, another important task takes place and consists of recognizing the category of each localized candidate. This task is known as object classification and is generally defined as a machine learning technique used to assign new instances or entries to a group of previously known instances called “class”. In our situation, this task of image classification permits the classification of image crops of ROIs based on their appearance features. These features are fed into an already trained classifier to recognize different object categories such as vehicles, persons, or animals. Therefore, this classification can be described as a combination of feature extraction and feature classification. Different feature extraction algorithms have been proposed in the literature and can be distinguished into two main categories: handcrafted feature extraction and machine-learned feature extraction. In the following, we will discuss in detail the processes of feature extraction and feature classification as the main parts of a traditional object classification task.

a. Feature extraction

Feature extraction is a popular topic in computer vision and refers to the process of transforming raw image pixels into a compact image representation ready for classification. There exist two main distinguished feature extractors: handcrafted features and machine-learned features. Handcrafted features are manually designed algorithms that extract representative characteristics from images. These extractors are founded on a prior knowledge and understanding of the targeted domain and usually involve techniques such as edge detection, corner detection, texture analysis, and color histograms. Performing efficient feature extraction is considered an important step in achieving efficient classification. Thus, developing efficient feature extraction algorithms has been a real challenge in computer vision for a long time and has been the subject of many studies since its earliest days. Feature extractors have evolved through various stages over the past few decades.

The first contributions in this context have focused on using global low-level image features such as color histograms or geometric shapes. For a long time, these approaches were considered robust due to their effectiveness and low computational complexity. However, despite their effectiveness, low-level images cannot be considered as strong distinctive features, especially when dealing with images that lack strong texture representation. Therefore, it makes them unable to fully satisfy the accuracy and consistency required in most situations [20], [21]. A second generation of extractors appeared as an alternative to previous ones which are known as local feature-based extractors. These methods consist of more sophisticated techniques to extract local low-level features such as corners and edges from images.

One of the earliest feature detector methods was proposed by Harris and Stephens [22] and is called the Harris corner detector. It is known for its powerful results even in noisy images where corner regions or patches are calculated using a shifted window in both directions. Despite its significant performance, the Harris detector is not considered invariant to scale and rotation transformations and requires defining specific thresholds for each image to operate correctly. In 2006, almost a decade after the Harris detector, a new corner detector algorithm called features from accelerated segment test (FAST) was introduced by Rosten and Drummond [23], and works by comparing the intensities of surrounding pixels of each candidate point to its intensity. This algorithm has the advantage of being computationally efficient and able to work in real-time applications. However, it is sensitive to noise and can produce false positives in regions with low texture [24]. Texture-based detectors represent the third generation of detectors proposed to overcome previous limitations. These extractors can extract key information from images even with the absence of well-defined edges, patterns, or textures. These detectors can be divided into two categories: local appearance-based methods and interest point-based methods. The first category includes methods such as local binary patterns (LBP) [25] and histogram of oriented gradient (HOG) [14], which operate by capturing local spatial patterns of image blocks based on the values of the pixels' neighbors. The second category consists of points of interest-based detectors that find distinguished key points of interest that are invariant to scaling, translation, rotation, and illumination changes. To match and retrieve these key points, several feature descriptors provide a compact encoded representation of the key points. This operation allows the encoding of essential information for image retrieval and comparison purposes. Many feature detectors and descriptors have been proposed in the literature including scale invariant feature transform (SIFT) [12], binary robust independent elementary features (BRIEF) [26], speeded-up robust features (SURF) [27], and oriented FAST and rotated BRIEF (ORB) [13].

b. Feature classification

One of the early approaches to feature classification was proposed by Csurka *et al.* [28] in 2004 and involved using a concept called a bag of key points to represent the image as a collection of key points, before quantizing them into a set of representative visual words. Afterward, a histogram of these words is computed to describe the feature vector of the image, which can be passed as input to a classifier for training or predicting tasks. Several classifiers might be used along with this approach to classify features such as support-vector machine (SVM), random forest, k-nearest neighbor (KNN), naïve Bayes, or AdaBoost, to train it. Other simpler approaches of feature classification operate by using low appearance representations such as image histograms of colors, and textures, or other feature descriptors such as local LBP or Gabor filters, as input for classification models. The effectiveness of these approaches depends on several factors, such as the quality and volume of the dataset, the quality and accuracy of detectors and descriptors algorithms, and the machine-learning algorithm used for classification. However, one of the main limitations of these traditional feature classifiers is the lack of spatial knowledge about processed images, which can lead to some information loss that could affect the performance.

2.2. Deep learning-based object detection

Due to their rapid growth and remarkable success, DCNNs have received a lot of attention in computer vision, especially in the domain of object classification and detection. Thus, since 2012, many studies have started adopting CNN models to address the problem of object detection, especially to overcome its persistent issues and limitations related to speed and accuracy in traditional detectors. A deep neural network is a system that attempts to mimic the biological neural network of the human brain through a combination of data inputs, weights, biases, and outputs [20]. Therefore, a CNN can be described as a specific type of deep neural network designed to process image pixels. CNN architectures have gained popularity in the computer vision field due to their good performance in terms of accuracy and speed compared to traditional approaches. This performance is due to the notable CNN capabilities that allow extracting efficiently deep image features at multiple scales and multiple levels. Moreover, the abundance and the accessibility to image datasets as well as to faster graphics processing units (GPUs), contributed greatly to the success of CNNs. Most CNN object detection models fall into two types: two-stage detectors and one-stage detectors [29]. Two-stage detectors refer to models that consist of two stages: i) a stage for region proposal generation used to locate candidate regions that could potentially contain objects and ii) and a second stage for proposal classification, which inspects and classifies each candidate region. On the other hand, single-stage architecture is considered an improved straightforward approach to detection compared to the previous one and operates by processing an image with a single pass through the CNN to generate bounding boxes and labels.

In both two-stage and single-stage detectors, the core of the detection process relies on a backbone network whose role is to extract image features using a stack of multiple convolutional layers, which are trained for this purpose. Since 2012, several backbone architectures have been proposed, among them there is AlexNet which is considered the first proposed CNN architecture for image classification in 2012 [30]. It is considered as one of the most influential contributions to computer vision due to its considerable accuracy compared to other contemporary models, which motivates researchers to switch their attention to CNN architectures. AlexNet architecture consists of eight neuron layers, including five convolutional and three fully connected layers, with up to 60 million parameters. It is considered the first architecture to use the non-saturating rectified linear unit (ReLU) activation function instead of tanh or sigmoid, allowing for considerable improvement in the training stage. On the other hand, VGG was developed as another powerful CNN architecture at the time in 2014 by Simonyan and Zisserman [31] based on AlexNet. VGG was the 1st runner-up in the classification task of the ImageNet large scale visual recognition challenge 2014 (ILSVRC2014) competition and achieved almost 92.7% accuracy on ImageNet [31]. Two variants of VGG have been proposed: VGG-16 and VGG-19. The first architecture variant contains a total of 16 layers, including 13 convolution layers for feature extraction and 3 fully connected layers for classification, while the VGG-19 has 16 convolution layers. The relatively large number of layers makes this network more capable with about 138 million parameters. VGG has shown that the capabilities of deep networks could be improved by increasing the number of parameters and the depth of network layers [30]. Another well-known CNN backbone architecture was residual neural network (ResNet) [32], which was one of the innovative approaches in deep learning. This architecture introduced new idea called "identity shortcut connection" which allows establishing skips and jumps over one or more layers instead of direct forward connections. This idea makes it possible to train and predict with larger and deeper neural networks while keeping a lower resource and time complexity.

2.2.1. Two-stage object detection

Two-stage detector ideas break detection into two tasks: a region proposal task performed by a component called a region proposal network (RPN), and a classification task to classify generated regions of interest that is usually done by a machine learning (ML) algorithm like SVM or a neural network-classifier. This approach represents the first generation of deep learning-based detectors and is known for being relatively slow but very accurate. Continuous progress in two-stage detectors has allowed for improving relatively the critical computational cost while maintaining good precision. Several two-stage detectors have been proposed in the literature, including R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN. R-CNN was proposed by Virasova *et al.* [33] in 2014. It was a pioneering work in deep learning detection that significantly improved performances and demonstrated the immense power of CNNs in object detection. As a first step, R-CNN uses the Selective Search algorithm instead of sliding windows to generate a set of around 2,000 region proposals, also called anchor boxes. In the second step, each proposal is resized to a fixed size and transformed into a fixed feature vector using a CNN feature extractor. Afterward, a trained SVM classifier is used to classify generated feature vectors to associate each anchor box with a label or class. Finally, a bounding box regression is performed by predicting four box coordinates, to refine anchor boxes and fit pre-classified objects. As a result of detection, R-CNN on PASCAL visual object classes challenge 2007 (PASCAL VOC2007) achieved a great score of 66% mAP, showing the huge advantage of this approach at the time. Despite this achievement, R-CNN has faced some critical limitations in practical applications, due to the heavily redundant computations that affect training and predicting time [34]. To accelerate detection, spatial pyramid pooling network (SPP-NET) [35] proposed an improved approach based on R-CNN that instead of applying the CNN feature extractor to each proposal separately, it extracted one global feature map from the entire image. Afterward, a fixed-length feature vector is obtained from the feature map for each proposal through the spatial pyramid pooling (SPP) layer. This strategy allows SPP-NET to achieve better performance compared to R-CNN, however, as it consists of multiple components like R-CNN, training an SPP-NET requires a lot of time.

In 2015, Fast R-CNN was proposed as an improved version of R-CNN by the same researchers while introducing some ideas from SPP-NET. To extract features, Fast R-CNN calculates the feature map of the image like SPP-NET before extracting feature vectors for each proposal using a region of interest pooling layer instead of a spatial pyramid pooling layer. After this step the feature vector layer is connected to two separate layers: a SoftMax layer for classification, and a regression layer for bounding box generation. These improvements make Fast R-CNN a near end-to-end model, except for the first region proposal stage which remains separate. This allows it to reach better training and prediction speed [29]. In the same year, Faster R-CNN was proposed by Ren *et al.* [36] with a main contribution of removing the Selective Search used in the first proposal stage that cannot be trained in a data-driven manner. This task is replaced with a novel trainable proposal generator called RPN. RPN is a fully convolutional network that generates a set of object proposal vectors on each anchor using different-sized sliding windows. These feature vectors allow for predicting the class score and locations of objects. In general, two-stage detectors have evolved over several years to improve detection performance, however, several major limitations remain challenging for these types of architectures, which are mainly related to their complexity and affect the speed of training and predicting.

2.2.2. One-stage object detection

On the other hand, to overcome the intrinsic limitations of two-stage detectors, one-stage detectors have been proposed as an end-to-end architecture of detection that can be trained and predicted directly via one pass by a single neural network. In general, these detectors predict different object bounding boxes and labels using a neural network that takes an image as input, without the need for any separate region proposal or feature extractor step. The main idea of one-stage detection usually starts by considering all regions on the image as potential objects and then attempting to classify each region as either a background or a target object. This straightforward approach is generally faster and more computationally efficient than the previous approach. However, despite their remarkable efficiency, the overall accuracy and precision of these detectors can be relatively modest, especially in the case of dealing with small or occluded objects. Compared to previous approaches to detection, the continuous development in recent years has allowed one-stage detectors to meet or even outperform the performance of two-stage detectors in terms of precision and accuracy in the majority of challenging detection situations. This enables the possibility of involving robust object detection in various real-time applications including autonomous driving, surveillance, and robotics. Among the relevant one-stage detectors are YOLO was first introduced by Redmon *et al.* [37] and SSD [38]. YOLO is one of the well-known one-stage detectors, proposed for the first time in 2016 as a real-time detector that can predict objects within images via a fully end-to-end CNN. YOLO considers the detection as a simultaneous regression and classification problem. The regression task separates the objects spatially via bounding boxes, and the classification task evaluates the probability of the associations of these boxes to the predefined classes. YOLO first starts by dividing the input image into a grid of N×N cells. For each cell, several bounding boxes get predicted with their corresponding confidence scores that range between zero and one, which indicate the likelihood of object existence, in addition to a score of similarity confidence between the box and each of the predefined classes. In addition to being flexible and easy to train and deploy, the continuous improvement in YOLO architecture allows it to become much faster and more accurate over time, and become more suitable and popular for real-time applications. However, despite these continuous improvements, in general, YOLO detectors still face some limitations regarding object localization tasks, which is due essentially to the basic grid-based approach that allows only a specific number of anchors to inspect for each bounding box. This causes the detector to struggle with cluttered situations, objects that occupy multiple cells, and partially occluded objects.

3. DEEP LEARNING DETECTORS FOR TRAFFIC SURVEILLANCE SYSTEMS

Due to the great success of deep learning in object detection, several studies have adopted deep learning detectors into their solutions to address current real-world problems such as traffic surveillance. Table 1 presents some of these recent studies that propose solutions for the challenging vehicle detection task using convolutional neural network (CNN) approaches. In the following, we present some of these approaches.

Table 1. Relevant works based on traditional or deep CNN detectors proposed for vehicle detection

		Traditional detectors												Deep learning detectors	
		Object localization methods			Feature extractors						Classifiers				
Paper	Year	Optical flow	Frame differencing	Background subtraction	Haar	CNN features	SIFT	HOG	ORB	Neural network	Random forest	AdaBoost	SVM	Single-stage detector	Two-stage detector
[6]	2011							×					×		
[39]	2012			×				×					×		
[7]	2012				×			×				×			
[40]	2013	×													
[8]	2014						×						×		
[1]	2015			×											
[2]	2016		×				×								
[15]	2017													Optimized YOLO	
[3]	2017			×				×					×		
[4]	2018			×		×				×					
[41]	2018					×						×	×		
[5]	2018	×	×												
[11]	2018			×											
[42]	2019													SSD/Mobile SSD	
[16]	2019								×					YOLOv3	
[43]	2019			×											
[17]	2019														Mask R-CNN
[9]	2019				×					×					
[44]	2019														Faster R-CNN
[45]	2020													Improved YOLOv3	
[10]	2020			×		×				×					
[18]	2021							×			×	×	×		
[46]	2021													YOLOv3	
[47]	2021													Improved YOLO	
[48]	2021													Tiny-YOLOv3	
[49]	2022													YOLOv4-Tiny	
[50]	2022													Improved YOLO4	
[51]	2022			×										YOLOv5	
[52]	2021					×									

Azimjonov and Özmen [47] proposes an improved YOLO-based vehicle detector intended for traffic flow monitoring systems. To further improve YOLO’s classification capability, the authors trained and tested several machine learning classifiers on 7,216 annotated images to choose the best among them. Finally, an improved YOLO-based detector was developed by combining the YOLO model, used in this case an object localizer, with the chosen classifier, which allows achieving an increase in classification precision up to 95.45% compared to the original 57% of ordinary YOLO. To capture traffic violations, Tajar *et al.* [48] proposed a lightweight real-time vehicle detection model developed for common and low-power systems. This model was based on a modified tiny-YOLOv3 model pruned and simplified by training it on a BIT-vehicle dataset. Experimental results showed a 95.05% mean average precision (mAP) while detecting six different types of vehicles at a speed of 17 fps. To tackle the problem of missing vehicles, Mao *et al.* [45] proposed an improved detector based on YOLOv3. This detector involved adding three spatial pyramid pooling layers (SPP) before each network layer to obtain enhanced multi-scale information, in addition to adopting soft-NMS instead of non-maximal suppression (NMS) to reduce missing bounding boxes. Results showed that this approach significantly reduces missed detection caused by overlapping vehicles in traffic scenes. Song *et al.* [16] proposes an improved system for vehicle counting based on YOLOv3 that involves dividing the highway road surface into a remote and a proximal area before introducing them to the detector. This approach allows for better vehicle localization, especially for small ones located in the remote area. After detection, ORB feature extraction is applied to vehicle regions to keep track of them over time using feature matching. In studies [15], an optimized YOLO is proposed to improve detections by replacing the last two fully-connected layers of the YOLO network with an average pool layer. This modification is motivated by the fact that fully-connected layers of CNNs that play the role of classifier require heavy computations. A combination of OYOLO with R-FCN [53] is proposed to avoid location errors in vehicle detection, allowing the mAP of OYOLO+R-FCN to increase by around 3.3% compared to YOLO. Chen *et al.* [42] proposes a real-time traffic density estimator that works by counting passing vehicles based on a MobileNet-SSD detector. The proposed detector was based on an SSD architecture trained on a handcrafted dataset of vehicles collected from traffic videos. Experiments showed that the proposed model was capable of handling vehicles with different scales and from different points of view, appearing in challenging lighting and weather conditions. The approach achieves an average detection accuracy of 79.30% and a detection speed between 15-18 fps, compared to the original SSD detector that achieved a higher accuracy of 92.97% but with a much slower rate that remains between 5-7 fps. Despite the notable efficiency achieved by the previous propositions, another kind of approach that has been around since the first appearance of CNN architectures until today has been used to enhance the detection performance of vehicles in traffic scenes. This involves adopting hybrid approaches that combine traditional computer vision algorithms and deep learning models. In this regard, Wang *et al.* [10] combines a mixture of the Gaussian background subtraction model (MOG2) with a tiny classification model called H-SqueezeNet to create a robust object detector for traffic surveillance systems. H-SqueezeNet is a lightweight architecture created by concatenating some SqueezeNet layers and trained on vehicle images from the ImageNet dataset. The idea behind this approach is to exploit MOG2 as a lightweight candidate proposal before classifying its results using H-SqueezeNet. Experiments showed that this approach can easily achieve real-time detection with a speed of 39.1 fps. Likewise, [51] proposes a vehicle detector system that operates based on a combination of background subtraction (BS) and YOLOv5. The Gaussian mixture model (GMM) is the core of the BS used as a preprocessing step to locate moving objects. The YOLOv5 detector is applied afterward to these regions separately to detect these moving regions. The result shows that this approach can efficiently handle vehicle overlapping problems. Different versions of YOLOv5 have been tested with this configuration. The smallest version, YOLOv5s, achieved a precision of 38.03% mAP with an inference speed of 52 ms per image, equivalent to 19 fps for almost real-time detection. The larger model, YOLOv5x, achieved a precision of 40.79% mAP with a speed of 263 ms or 3.8 fps. Numerous approaches have been proposed for vehicle detection, utilizing both traditional and deep convolutional neural network (CNN) detectors. Table 1 summarizes relevant works in this area, highlighting the different methods and techniques used to address traffic surveillance problem. This table provides a comprehensive comparison of various techniques, highlighting the increasing tendency to adopt deep learning techniques in recent works, particularly one-stage detectors, in contemporary traffic surveillance applications.

4. EXPERIMENTAL AND RESULTS

In general, comparing the CNN detectors can be a challenging task due to the variety of factors that may influence their performance and evaluation. These factors include the dataset quality and the evaluation metric, in addition to the hardware performance used for this evaluation. Several open image datasets have been proposed to facilitate the tasks and enable meaningful evaluation. These datasets include PASCAL VOC2007/2012, ImageNet, ILSVRC, and Microsoft COCO datasets. Recently common objects in context (COCO) have been considered as a popular large-scale image dataset, intended to evaluate classification, detection, and segmentation tasks. It provides over 330k images containing more than 1.5 million object instances across over 80 object categories. COCO is widely used in object detection due to its large and diverse collection of accurately annotated real-world images. Its evaluation metrics are based on Mean average precision (mAP), which measures the precision and accuracy of object detections at multiple intersection over union (IoU) thresholds.

4.1. Experimental conditions and results

In our experiments, we have compared some of the recent well-known one-stage detectors, in terms of detection speed and precision as shown in Figure 2. In recent years, two-stage detectors have been largely outperformed by one-stage detectors in terms of processing speed due to their straightforward architectures. For this reason, we have decided our experiment to focus only on evaluating one-stage detectors due to their popularity and their remarkable performances. Hence, the chosen detectors are YOLO models from version 3 to the recent version to date which is 8, and with different model sizes that vary from small to X large models. In general, to evaluate the computational efficiency of CNN-based models, traditional methods such as big-O notation for computational complexity become not useful. Instead, inference speed is considered a popular metric to indicate computational efficiency. Another popular metric for efficiency evaluation is the processing rate revealed by the number of frames per second (FPS). Hence, to evaluate our detectors regarding processing speed, we measure and represent the FPS achieved by each model in detecting objects within the frames of an experimental traffic sequence as shown in the x-axis of Figure 2. The FPS in our experiment represents the number of frames in the sequence that a detector can process in one second, bearing in mind that the higher the FPS, the more efficient the detector is. The experimental sequence is a highway traffic video of about five minutes with a total of 6,720 frames that have a resolution of 1280×720. The FPS calculation was performed for each one of the compared detectors using a local machine with 32 GB RAM, an Intel Xeon Silver 4,110-2.10 GHz processor, and no integrated graphics card. The models in our experiment were trained on the default COCO dataset, and all detections were performed using a network input size of 416×416. On the other hand, concerning precision performances shown in the Y-axes of Figure 2, the MAP scores of our evaluated detectors are collected from the table of performances published in their original papers as in Table 2.

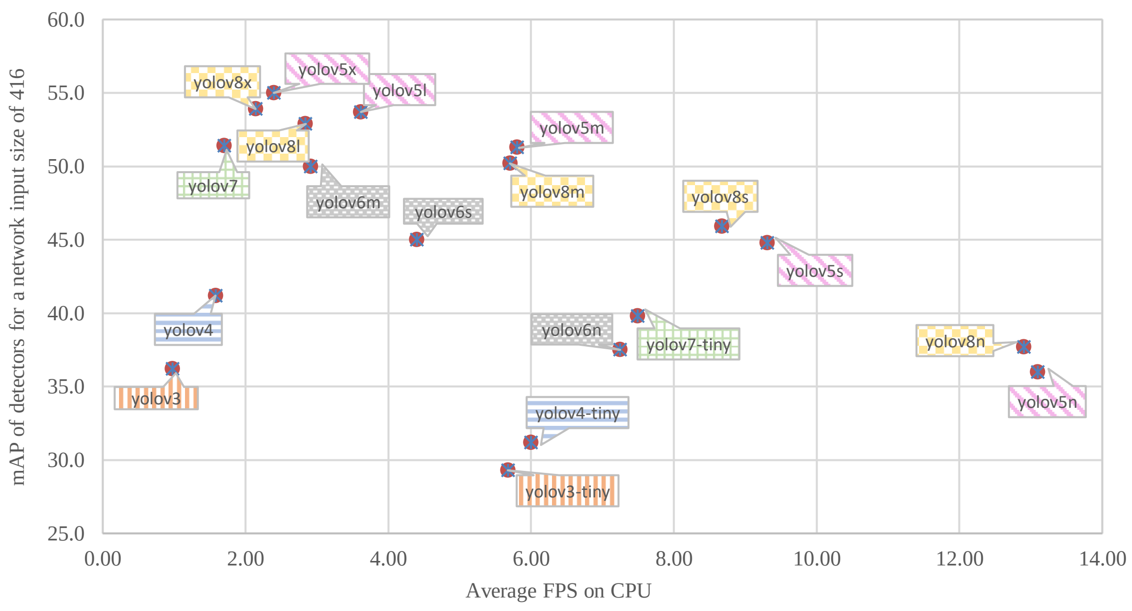

Figure 2. A comparison between evaluated object detectors in terms of FPS and accuracy

Table 2. The performance of evaluated detectors

Model (with 416 input)	Processing time (ms)	FPS	AP
YOLOv3	1020,83	0,98	36,2
YOLOv3-tiny	176,28	5,67	29,3
YOLOv4	633,07	1,58	41,2
YOLOv4-tiny	166,67	6,00	31,2
YOLOv5n	76,36	13,10	36,0
YOLOv5s	107,47	9,31	44,8
YOLOv5m	172,29	5,80	51,3
YOLOv5l	276,58	3,62	53,7
YOLOv5x	417,48	2,40	55,0
YOLOv6n	137,99	7,25	37,5
YOLOv6s	227,49	4,40	45,0
YOLOv6m	343,50	2,91	50,0
YOLOv7-tiny	133,54	7,49	39,8
YOLOv7	587,01	1,70	51,4
YOLOv8n	77,53	12,90	37,7
YOLOv8s	115,27	8,68	45,9
YOLOv8m	175,38	5,70	50,2
YOLOv8l	352,56	2,84	52,9
YOLOv8x	467,13	2,14	53,9

The overall results show that the precision of compared models generally varies according to the size of the models, where the precision falls between 30 and 55 mAP, and their speed varies between 1 and 13 fps. In general, by looking at the relation between model sizes and their performances, we observe that by going larger, the detector becomes more accurate but at the same time its speed becomes much slower, and vice versa. Moreover, regarding the version of models and their performances, the continuous development and optimization of YOLO architectures over time allows them to become faster and more accurate according to the same results. Another important feature of recent versions of YOLO is the tendency to release multiple model versions with different network sizes that enable multiple accuracy/speed detection ratios. This variety allows developers to choose the optimal model regarding the targeted use case, and also the targeted physical resource of deployment. For Instance, recent larger networks such as models X and L of

YOLOv5 and YOLOv8 in addition to YOLOv7 tend to be more precise with an AP of more than 50%, compared to early, medium, and small detectors. However, the achieved speed of these precise detectors is considerably slow with a rate below 4 fps on CPU as shown in the result of our experiment, which means that these models are far from being effective for real-time applications, especially while operating in CPU machines or low-resource devices. On the other hand, in terms of speed, results show a large advantage of recent nano versions of YOLO models, such as YOLOv8 nano and YOLOv5 nano. These models achieve the best speed performance in the CPU, with near real-time detection with more than 13 fps. However, despite their remarkable speed, their reduced size affects largely their detecting capability, which is due especially to the lack of the ability to detect small and occluded objects. Whereas small recent models of YOLOv8 and YOLOv5 have around 3 times the number of parameters of Nano versions, they achieve around 9 fps for both of them, with more remarkable detecting capability compared to previous ones.

5. CONCLUSION

In this paper, we conducted a literature review of recent vision-based studies that address the problem of vehicle detection in traffic surveillance systems. We began with an overall review of object detection approaches including relevant handcrafted-based and deep learning-based approaches. We also highlighted some relevant studies proposed to address the problem of vehicle detection. In the last section, we provide a comparative analysis of various well-known one-stage detectors based on their speed and accuracy performances while dealing with real-world traffic surveillance footage. The results showed that one-stage detectors, particularly recently reduced YOLO networks, have become more efficient with an acceptable accuracy/speed ratio. In particular, as shown in Table 1 the small and nano models of YOLOv5 and YOLOv8 demonstrate superior performance among the evaluated models, with small models excelling in precision and nano models in speed. This dual advantage offers the possibility to develop hybrid approaches that adaptively employs both models, dynamically switching between them based on the specific demands of the traffic surveillance task, thereby optimizing both accuracy and real-time processing capabilities.

REFERENCES

[1] G. Lee, R. Mallipeddi, G. J. Jang, and M. Lee, "A genetic algorithm-based moving object detection for real-time traffic surveillance," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1619–1622, 2015, doi: 10.1109/LSP.2015.2417592.

[2] K. Mu, F. Hui, and X. Zhao, "Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching," *Journal of Information Processing Systems*, vol. 12, no. 2, pp. 183–195, 2016, doi: 10.3745/JIPS.02.0040.

[3] Wahyono and K. H. Jo, "Cumulative dual foreground differences for illegally parked vehicles detection," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2464–2473, 2017, doi: 10.1109/TII.2017.2665584.

[4] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Vision-based occlusion handling and vehicle classification for traffic surveillance systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 80–92, 2018, doi: 10.1109/MITS.2018.2806619.

[5] Y. Peng, Z. Chen, Q. M. J. Wu, and C. Liu, "Traffic flow detection and statistics via improved optical flow and connected region analysis," *Signal, Image and Video Processing*, vol. 12, no. 1, pp. 99–105, 2018, doi: 10.1007/s11760-017-1135-2.

[6] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 2421–2424, doi: 10.1109/ICIP.2011.6116132.

[7] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2012, pp. 951–956, doi: 10.1109/ITSC.2012.6338852.

[8] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635–1647, 2014, doi: 10.1109/TGRS.2013.2253108.

[9] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, "Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 54–64, Jan. 2019, doi: 10.1109/TITS.2018.2797697.

[10] Z. Wang, J. Huang, N. N. Xiong, X. Zhou, X. Lin, and T. L. Ward, "A robust vehicle detection scheme for intelligent traffic surveillance systems in smart cities," *IEEE Access*, vol. 8, pp. 139299–139312, 2020, doi: 10.1109/ACCESS.2020.3012995.

[11] M. Anandhalli and V. P. Baligar, "An approach to detect vehicles in multiple climatic conditions using the corner point approach," *Journal of Intelligent Systems*, vol. 27, no. 3, pp. 363–376, 2018, doi: 10.1515/jisys-2016-0073.

[12] Lowe D.G, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571, doi: 10.1109/ICCV.2011.6126544.

[14] D. Navneet and T. Bill, "Histograms of oriented gradients for human detection," in *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2020, vol. 16, no. 7, pp. 4714–4725.

[15] J. Tao, H. Wang, X. Zhang, X. Li, and H. Yang, "An object detection system based on YOLO in traffic scene," in *Proceedings of 2017 6th International Conference on Computer Science and Network Technology, ICCSNT 2017*, 2017, vol. 2018-Janua, pp. 315–319, doi: 10.1109/ICCSNT.2017.8343709.

[16] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review*, vol. 11, no. 1, 2019, doi: 10.1186/s12544-019-0390-4.

[17] E. P. Ijjina, D. Chand, S. Gupta, and K. Goutham, "Computer vision-based accident detection in traffic surveillance," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2019, pp. 1–6, doi: 10.1109/ICCCNT45670.2019.8944469.

[18] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing and Applications*, vol. 33, no. 2, pp. 613–624, 2021, doi: 10.1007/s00521-020-05002-6.

[19] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.

[20] A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and Y. J. Patel, "A review: Object detection models," *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, 2021, pp. 1-8, doi: 10.1109/I2CT51068.2021.9417895.

[21] M. M. El-Gayar, H. Soliman, and N. Meky, "A comparative study of image low level feature extraction algorithms," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 175–181, 2013, doi: 10.1016/j.eij.2013.06.003.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference 1988*, 1988, pp. 23.1--23.6, doi: 10.5244/C.2.23.

[23] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 430–443, 2006, doi: 10.1007/11744023_34.

[24] T. Georgiou, Y. Liu, W. Chen, and M. Lew, "A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision," *International Journal of Multimedia Information Retrieval*, vol. 9, no. 3, pp. 135–170, 2020, doi: 10.1007/s13735-019-00183-w.

[25] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006, doi: 10.1109/TPAMI.2006.244.

[26] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6314, no. 4, 2010, pp. 778–792.

[27] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: Speeded up robust features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951, 2006, pp. 404–417.

[28] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision*, vol. 1, no. 1, pp. 1–2, 2004.

[29] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing: A Review Journal*, vol. 126, Jun. 2022, doi: 10.1016/j.dsp.2022.103514.

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

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arxiv.org/abs/1409.1556*, Sep. 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[33] A. Y. Virasova, D. I. Klimov, O. E. Khromov, I. R. Gubaidullin, and V. V Oreshko, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Radioengineering*, pp. 115–126, 2021, doi: 10.18127/j00338486-202109-11.

[34] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," *Computer Vision and Pattern Recognition*, vol. 14, no. 2, pp. 53–57, 2009, doi: 1704.04861.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, no. PART 3, 2014, pp. 346–361.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137-1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[38] W. Liu *et al.*, "SSD: single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 21–37.

[39] X. Shi, H. Ling, E. Blasch, and W. Hu, "Context-driven moving vehicle detection in wide area motion imagery," in *Proceedings - International Conference on Pattern Recognition*, 2012, pp. 2512–2515.

[40] S. Aslani and H. Mahdavi-Nasab, "Optical flow based moving object detection and tracking for traffic surveillance," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 9, pp. 761–765, 2013.

[41] W. Chen, Q. Sun, J. Wang, J. J. Dong, and C. Xu, "A novel model based on AdaBoost and deep CNN for vehicle classification," *IEEE Access*, vol. 6, pp. 60445–60455, 2018, doi: 10.1109/ACCESS.2018.2875525.

[42] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using single shot detection (SSD) and MobileNet-SSD," *Physics and Chemistry of the Earth*, vol. 110, pp. 176–184, 2019, doi: 10.1016/j.pce.2018.12.001.

[43] Y. Wang *et al.*, "Detection and classification of moving vehicle from video using multiple spatio-temporal features," *IEEE Access*, vol. 7, pp. 80287–80299, 2019, doi: 10.1109/ACCESS.2019.2923199.

[44] A. Mhalla, T. Chateau, S. Gazzah, and N. E. Ben Amara, "An embedded computer-vision system for multi-object detection in traffic surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4006–4018, 2019, doi: 10.1109/TITS.2018.2876614.

[45] Q. C. Mao, H. M. Sun, L. Q. Zuo, and R. S. Jia, "Finding every car: a traffic surveillance multi-scale vehicle object detection method," *Applied Intelligence*, vol. 50, no. 10, pp. 3125–3136, 2020, doi: 10.1007/s10489-020-01704-5.

[46] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021, doi: 10.1109/TITS.2020.3025687.

[47] J. Azimjonov and A. Özmen, "A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways," *Advanced Engineering Informatics*, vol. 50, p. 101393, Oct. 2021, doi: 10.1016/j.aei.2021.101393.

[48] A. Taheri Tajar, A. Ramazani, and M. Mansoorizadeh, "A lightweight Tiny-YOLOv3 vehicle detection approach," *Journal of Real-Time Image Processing*, vol. 18, no. 6, pp. 2389–2401, 2021, doi: 10.1007/s11554-021-01131-w.

[49] X. Xiang, F. Meng, N. Lv, and H. Yin, "Engineering vehicles detection for warehouse surveillance system based on modified YOLOv4-tiny," *Neural Processing Letters*, vol. 55, no. 3, pp. 2743–2759, 2023, doi: 10.1007/s11063-022-10982-8.

[50] J. Zhao *et al.*, "Improved vision-based vehicle detection and classification by optimized YOLOv4," *IEEE Access*, vol. 10, pp. 8590–8603, 2022, doi: 10.1109/ACCESS.2022.3143365.

[51] Z. Charouh, A. Ezzouhri, M. Ghogho, and Z. Guennoun, "A resource-efficient CNN-based method for moving vehicle detection," *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22031193.

[52] M. Anandhalli, V. P. Baligar, P. Baligar, P. Deepsir, and M. Iti, “Vehicle detection and tracking for traffic management,” *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 66–73, 2021, doi: 10.11591/ijai.v10.i1.pp66-73.
[53] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: object detection via region-based fully convolutional networks,” *Advances in neural information processing systems*, vol. 29, pp. 379–387, 2016.

BIOGRAPHIES OF AUTHORS

Ayoub El-Alami received his master’s degree in business intelligence engineering from the FST of Moulay Slimane University, Morocco, in 2015. Currently, He is a Ph.D. student in the ENSET of Mohammedia, an affiliate of UH2C, Morocco. His research interests include computer vision, and deep learning and CNN-based approaches of object detection and tracking, in addition to real world application of these approaches, such as vehicle detection and intelligent traffic surveillance systems. He can be contacted at email: ayoubalami6@gmail.com.

Younes Nadir received his Ph.D. degree in engineering sciences at the National Higher School of Electricity and Mechanics (ENSEM), Hassan II University of Casablanca and is a professor at the National Higher School of Art and Design (ENSAD), Hassan II University of Casablanca. He is now responsible of the interactive and graphic design (DGI) training program at the same establishment. His research interests are mainly in computer vision, and artificial intelligence. He can be contacted at email: younes.nadir@ensad.ma or nadir.younees@gmail.com.

Khalifa Mansouri was born in 1968 in Azilal, Morocco. He is currently a teacher-researcher in computer science, deputy director in charge of training and director of the M2S2I Research Laboratory at ENSET of Mohammedia, Hassan II University of Casablanca. His research interests include information systems, e-learning systems, real time systems, industrial systems (modeling, optimization, and numerical computation) and artificial intelligence. Graduated from ENSET Mohammedia in 1991, CEA in 1992 and Ph.D. in computation and optimization of structures in 1994, HDR in 2010 and National Ph.D. in computer science in 2016. He is the author of 10 books in computer science, a scientific book with the publisher Springer, 422 research papers including 227 in the Scopus library and supervised 35 defended doctoral theses. He can be contacted at email: khalifa.mansouri@enset-media.ac.ma.