IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2021, accepted November 30, 2021, date of publication December 7, 2021,
date of current version December 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133529

Multiple Hypothesis Detection and Tracking Using Deep Learning for Video Traffic Surveillance

HAMD AIT ABDELALI[1], HATIM DERROUZ[1,2], (Member, IEEE), YAHYA ZENNAYI[1], RACHID OULAD HAJ THAMI[2], AND FRANÇOIS BOURZEIX[1]

[1]Embedded System and AI Department, MAScIR, Rabat 10100, Morocco
[2]IRDA Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Université Mohammed V de Rabat, Rabat 10100, Morocco

Corresponding authors: Hamd Ait Abdelali (h.aitabdelali@mascir.ma) and Hatim Derrouz (h.derrouz@ieee.org)

This work was supported in part by the National Center for Scientific and Technical Research (CNRST), and in part by the Ministry of Higher Education, Scientific Research and Executive Training (MESRSFC), through the Development of an Integrated System for Traffic Management and Detection of Road Traffic Infractions Project.

ABSTRACT Moroccan Intelligent Transport System is the first Moroccan system that uses the latest advances in computer vision, machine learning and deep learning techniques to manage Moroccan traffic and road violations. In this paper, we propose a fully automatic approach to Multiple Hypothesis Detection and Tracking (MHDT) for video traffic surveillance. The proposed framework combines Kalman filter and data association-based tracking methods using YOLO detection approach, to robustly track vehicles in complex traffic surveillance scenes. Experimental results demonstrate that the proposed approach is robust to detect and track the trajectory of the vehicles in different situations such as scale variation, stopped vehicles, rotation, varying illumination and occlusion. The proposed approach shows a competitive results (detection: 94.10% accuracy, tracking: 92.50% accuracy) compared to the state-of-the-art approaches.

INDEX TERMS Traffic surveillance, computer vision, deep learning, Kalman filter, data association, detection, multiple hypotheses tracking, occlusion handling.

I. INTRODUCTION

In the recent past [1]–[4], the interest of many researchers has been captured by the deployment of automated systems for video traffic surveillance. The change of illumination causes the most challenging factors in video traffic surveillance, deformation of vehicles, pause, motion blur, occlusions, and camera view angle, etc. Although traffic surveillance has been studied for several decades and numerous methods have been proposed for different tasks [5]–[8], it remains to be a very challenging problem. In the literature, traffic surveillance methods can be divided into two approaches, online model and offline models. An online model is a hard problem that receives video sequence input on a frame-by-frame basis and has to give an output for each frame. Offline models, allows for global optimization of the path, scanning forwards and backwards through the frames of a video sequence. Since offline models have access to more information, better performance is expected from these models.

Furthermore, there is no single method that can be successfully applied to all tasks and situations. However, recent progress on Multiple Object Tracking (MOT) has focused on the tracking-by-detection strategy, to solve the ambiguities in associating objects detection and to overcome detection failures. The majority of recent works process video sequence in a batch mode in which video frames from future time steps are also used to solve the data association problem. The general idea is to first localize, for each frame, all objects using an object detector, then associate the detected objects between frames using features such as location and appearance. A common methodology is to split the tracking into two phases: prediction of object location, and matching of detections and predictions. That is, for each new frame, the complete tracking model does the following: detect objects of interest, predict the new locations of the detected objects from previous frames, associate the detected objects between frames by the similarity of detected and predicted locations.

This paper describes a framework for detection and tracking of multiple vehicles using video sequences from a surveillance camera. In this paper, we will focus our attention on Multiple Hypotheses Detection and Tracking (MHDT) for video traffic surveillance. The proposed framework combines Kalman filter and data association-based tracking methods using YOLO detection approach [9]. Our result show that the

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

proposed method is robust to detect and track the trajectory of the vehicles in different situations (scale variation, pause, rotation, and occlusion).

The rest of the paper is organized as follows: Section 2 presents the Related work and motivation. Section 3 presents the problem formulation and methodology. Section 4 presents the System Overview. Section 5 presents the experiments results and Section 6 concludes the paper.

II. RELATED WORK AND MOTIVATION

In this part, we will provide an overview of the state-of-the-art techniques for detection and Multiple Objects Tracking (MOT) [5]–[8], [10]–[12]. Multiple Object Tracking is one of the earliest successful algorithms for visual tracking. Originally proposed in 1979 by Reid [13], it builds a tree of potential track hypotheses for each candidate target, thereby providing a systematic solution to the data association problem. However, this method is limited when it comes to modelling the complex appearance changes of a target. Prokaj and al [14], [15] presented a technique for Multiple objects tracking in an aerial road surveillance network. The moving object detection was performed using background subtraction, where the background was displayed as the method of a stabilized sliding window of frames. At that point, the information association problem was detailed as induction in a lot of Bayesian networks using movement and appearance consistency. This methodology avoided the exhaustive evaluation of data association hypotheses. Andriyenko and al. [8] proposed a discrete continuous optimization technique to solve trajectory estimation and data association, these two stages are exchanged until convergence. In [10], a fixed appearance model incorporated into a standard MOT system [16]–[18]. Interestingly, MOT can be extended to include online learned discriminative appearance models for each track hypothesis.

Currently, deep neural networks have been employed in multiple objects tracking frameworks [19]–[27], but unfortunately, highly accurate tracking algorithms based on CNN are often too slow for practical systems. MDNet [20] is a well known CNN-based tracking method with state-of-the-art accuracy. This method is inspired by an object detection network, R-CNN [28]–[30], it samples candidate areas, which are passed through a CNN pre-trained on a huge scale dataset and fine-tuned at the first frame in a test video sequence. Since each candidate is processed independently, MDNet suffers from high computational complexity in terms of time and space. In addition, while its multi-domain learning framework concentrates on the saliency of the target against the background in each domain, it is not optimized to distinguish potential target instances across multiple domains. Consequently, a learned model by MDNet is not the optimal choice to distinguish represent unseen target objects with similar semantics in test sequences. In [31] Abdelwahab and al, presented a rapid and reliable traffic congestion detection method based on robust texture and motion features extracted from traffic videos. In [32] Abdelwahab and al, presented a temporal pooling method to generate a dynamic image descriptor using a deep residual neural network to extract texture Features. The Table 1 showing the works reviewed in terms of its main distinguishing features.

TABLE 1. The reviewed works according to their main distinguishing features.

Algorithms	Features	Occlusion	Classifier	Tracking	Dataset	Accuracy
[5]	HOG	Low	MIL	OMRC	OTB2015	67.80%
[6]	RFD	Low	Boosting	BAFMPL-MOT	CAVIAR	91%
[19]	CNN	Low	CNN	CNNs	VOT2014	60%
[20]	CNN	Low	CNN	CNNs	VOT2015	93.70%
[21]	DCF	Medium	N/A	DCF trackers	UAV123	65%
[22]	DCF	Low	N/A	DCF methods	OTB-2015	82.40%
[23]	CNN	High	CNN	DeepSRDCF	OTB-50	56.50%
[31]	LTR	High	SVM	N/A	UCSD	97.64%
[24]	CNN	Low	CNNs	FCNT	OOT2014	56.65%
[32]	CNN	High	SVM	N/A	UCSD, NU1	97.64%

As noted early on, there is no single tracking method that can be successfully applied to all tasks and situations such as scale variation, pause, rotation, lighting variation, and occlusion. In this paper, our overall goal is to develop a fully automatic approach that can detect and track different vehicles and record their trajectories.

III. PROBLEM FORMULATION AND METHODOLOGY

A. PROBLEM FORMULATION

We will start by endeavoring to give a general mathematical formulation of MOT, discuss its possible categorizations based on different aspects. Let i be a vehicle appearing in a frame t. Let δ be a binary function that indicate the presence of the vehicle i in the frame t, where $\delta_t^i = 1$ if i apperas and $\delta_t^i = 0$ otherwise. The state of the vehicle i in the frame t is represented as $x_t^i = (p_t^i, w_t^i, h_t^i, v_t^i)$, where $p_t^i = (p_t^i(x), p_t^i(y))$ is the vehicle i center location, w_t^i and h_t^i are the width and height of its bounding box, and $v_t^i = (v_t^i(x), v_t^i(y))$ represents its velocity. We then define the track T_t^i of the vehicle i as a set of states up to frame t and denote it as $T_t^i = \left\{x_k^i | \delta_t^i = 1 \leqslant t_s^i \leqslant t_e^i \leqslant t\right\}$, where t_s^i and t_e^i are the start-frame and end-frame of the tracks, respectively. In addition, $T_{1:t}^i = \left\{T_1^i, T_2^i, \cdots, T_n^i\right\}$ are the states of all the n vehicles in the t-th frame, and $T_{1:t} = \left\{T_{1:t}^1, T_{1:t}^2, \ldots, T_{1:t}^n\right\}$ is the set of tracks of all the n vehicles up to frame t. Correspondingly, $d_t^j = (p_d, w_d, h_d)_t^j$ is the j-th detected observation at frame t, with p_d being the position of the centre location (given by its coordinates $(p(x), p(y))$), and w_d and h_d being the width and the height, respectively, of the detected vehicle. We also define $D_t = \left\{d_t^j; 1 \leqslant j \leqslant n\right\}$ as the set of the n detected vehicles (observations) at frame t. All the observations associated with vehicle i up to frame t are referred to as $d_{1:t}^i = \left\{d_1^i, \ldots, d_t^i\right\}$, and $D_{1:t} = \left\{d_{1:t}^1, \ldots, d_{1:t}^n\right\}$ is the set of all observations up to frame t. Given the set of trees that contains all trajectory hypotheses (Figure 1) for all targets, we want to determine the most likely combination of vehicle tracks at frame t. This can be formulated as a k-dimensional assignment problem: $\max_z \sum_{i_1=0}^{D_1} \sum_{i_2=0}^{D_2} \cdots \sum_{i_t=0}^{D_n} s_{i_1 i_2 \ldots i_n} z_{i_1 i_2 \ldots i_n}$ subject

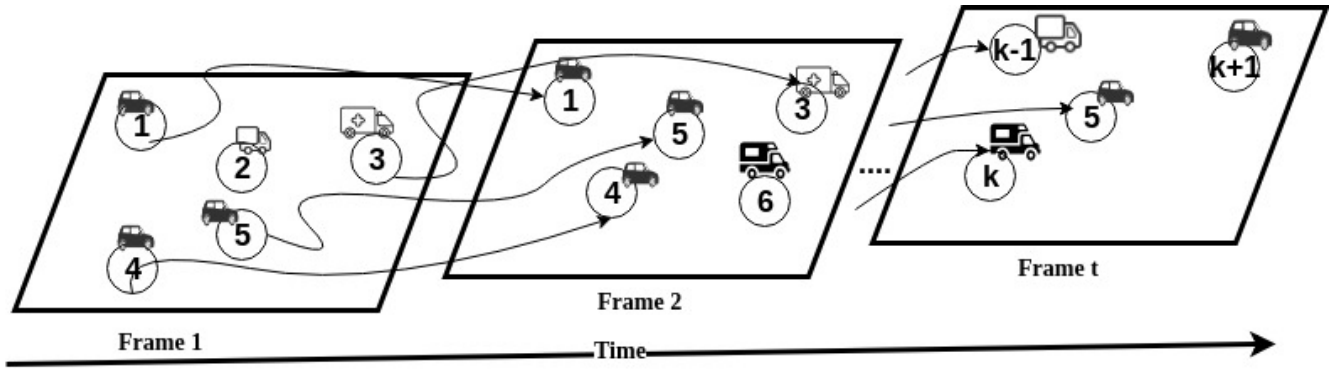


FIGURE 1. Track hypotheses after Detection stage.

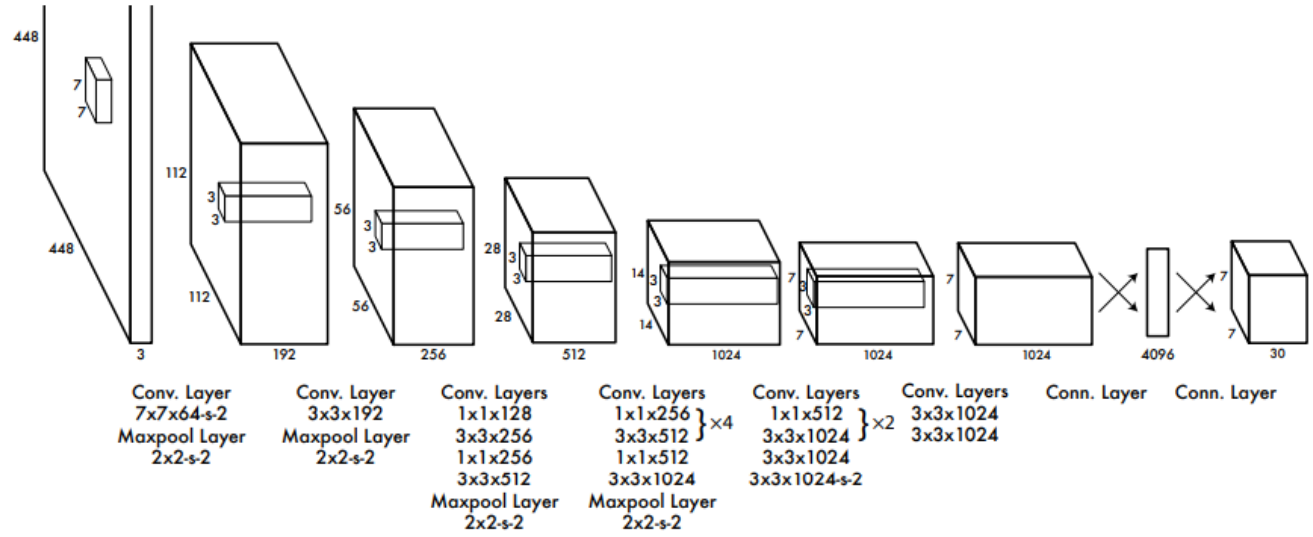


FIGURE 2. The YOLO convolutional neural network architecture [9].

to $\sum_{i_1=0}^{D_1} \sum_{i_2=0}^{D_2} \cdots \sum_{i_n=0}^{D_n} z_{i_1 i_2 \ldots i_n} = 1$ for $i_n = 1, 2, \ldots, D_n$. For each observation i we define one constraint to ensures that i_n is assigned to a unique track. Each track is associated with its binary variable $z_{i_1 i_2 \ldots i_n}$ and track score $s_{i_1 i_2 \ldots i_n}$. Thus, the objective function represents the total score of the tracks in the global hypothesis associations.

B. METHODOLOGY

The first step in our algorithm is to detect the vehicles. To do this, we begin by using the YOLO algorithm [9]. This algorithm based on two main steps to a predefined size of images during the learning process:

- Vehicle detection operated by convolution neural networks;
- A grid of the image that predicts the vehicle if it exists.

For detected vehicles, a tracking model is constructed using the Kalman filter. The algorithm is used to associate the temporally detected vehicles from one frame to the next. To do this, we use the Hungarian algorithm to associate data for a given frame, using the distance between the outputs of the algorithm and the Kalman filter estimation.

C. DEEP NEURAL NETWORKS FOR VEHICLES DETECTION

YOLO is implemented as a deep convolutional neural network. The open source implementation released along with the paper is built upon a custom Deep Neural Network (DNN) framework written by Redmon *et al.* [9]. YOLO reframes vehicle detection as a single regression problem, straight from image pixels to bounding box coordinates. YOLO divides the input image into a $S \times S$ grid. If the centre of a vehicle falls into a grid cell, that grid cell is responsible for detecting that vehicle. A description of the YOLO architecture can be seen in Figure 2:

D. KALMAN FILTER

A Kalman filter [33] is a recursive algorithm which predicts the state variables and further uses the observed data (measurements) to correct/update the predicted value. Kalman filter has two main states: the state prediction and the state correction.

1) STATE PREDICTION

For each time step t, a Kalman filter first makes a prediction $\hat{x_t}$ of the state:

$$\hat{x_t} = A \times x_{t-1}, \tag{1}$$

where x_{t-1} is a vector representing the process state at time step $t-1$ and A is a process transition matrix. The Kalman filter concludes the state prediction steps by projecting an estimate error covariance P_t^- forward one time step:

$$P_t^- = A \times P_{t-1} \times A^T + W, \tag{2}$$

where P_{t-1} is a matrix representing error covariance in the state prediction at time step $t-1$ and W is the process noise covariance (or the uncertainty in our model of the process).

2) STATE CORRECTION

After predicting the state $\hat{x_t}$ (and its error covariance) at time t using the state prediction steps, the Kalman filter uses measurements to "correct" its prediction during the measurement update steps. First, the Kalman filter computes a Kalman gain K_t, which is later used to correct the state estimate $\hat{x_t}$:

$$K_t = P_t^- \times (P_t^- + R_t)^{-1}, \tag{3}$$

where R_t is the noise covariance measurement. Determining R_t for a set of measurements is often difficult. In our contribution we calculated R_t dynamically from the measurement state.

Using Kalman gain K_t and z_t measurements from time step t, we can update the estimate state:

$$\hat{x_t} = \hat{x_t} + K_t \times (z_t - \hat{x_t}). \tag{4}$$

Conventionally, the measurements z_t are often derived from sensors. In our approach, the measurements z_t are the output of the tracking algorithm gave the same input: one frame of a streaming video and the most likely $p(x)$ and $p(y)$ coordinates of the target vehicle in this frame (taking the first two dimensions of $\hat{x_t}$).

The final step of the Kalman filter iteration is to update the error covariance P_t^- into P_t:

$$P_t = (I - K_t) \times P_t^-. \tag{5}$$

The updated error covariance will be significantly decreased if the measurements are accurate and slightly decreased if the measurements are noisy.

E. DATA ASSOCIATION

The data association are solved using Hungarian technique (also known as Kuhn-Munkres algorithm) [34]. This technique is used to associate the identified objects in frame t to the unidentified objects in frame $t+1$ by finding the extreme solution of the Bhattacharyya distance [35] in the assignment matrices.

The first association stage solves the assignment problem between the active tracks T_{t-1} and the current detections D_t to progressively build vehicle trajectories. The input pairs for this stage are $\left\{\left(T_{t-1}^i, d_t^j\right) \mid \forall T_{t-1}^i \in T_{t-1}, \forall d_t^j \in D_t\right\}$, and the association is evaluated using the following affinity model:

$$\mathbb{A}\left(T_{t-1}^i, d_t^j\right) = \mathbb{A}_a\left(T_{t-1}^i, d_t^j\right) \mathbb{A}_s\left(T_{t-1}^i, d_t^j\right) \mathbb{A}_m\left(T_{t-1}^i, d_t^j\right), \tag{6}$$

where $\mathbb{A}_a\left(T_{t-1}^i, d_t^j\right)$, $\mathbb{A}_s\left(T_{t-1}^i, d_t^j\right)$ and $\mathbb{A}_m\left(T_{t-1}^i, d_t^j\right)$ are the appearance affinity, shape affinity and motion affinity, respectively.

Let x_{d^j} be the bounding box of a detection d_t^j, x_{T^i} be the latest bounding box of the track T_{t-1}^i, and $H_{T^i} = \{x_{T^i}\}$ be the historical bounding box set of the track T_{i-1}^i, The Bhattacharyya distance [35] is used to evaluate the similarity between two templates, and define the appearance affinit $\mathbb{A}_a$ of the track T_{t-1}^i and the detection d_t^j as:

$$\mathbb{A}_a\left(T_{t-1}^i, d_t^i\right) = \Omega(T_{t-1}^i)\rho(x_T, d^j) + (1 - \Omega(T_{t-1}^i)) \max_k \rho(x_{T^i}, x_{d_j}), \tag{7}$$

where $\rho(.,.)$ is the Bhattacharyya distance, and $\Omega(T_{t-1}^i) \in [0, 1]$.

The shape affinity, $\mathbb{A}_s(T_{t-1}^i, d_t^j)$ (in Equation (6)), between the track and the detection is defined as:

$$\mathbb{A}_s(T_{t-1}^i, d_t^j) = exp\left(-\left\{\frac{h^i - h_d^j}{h^i + h_d^j} + \frac{w^i - w_d^j}{w^i + w_d^j}\right\}\right), \tag{8}$$

where (w^i, h^i) are the width and the height of the bounding box of the tail of track T_{t-1}^i and (w_d, h_d) are the width and the height of the bounding box of the detection d_t^j.

The motion affinity, $\mathbb{A}_m$ (in Equation (6)), is evaluated between the tail of the history of the track T_{t-1}^i and the detection d_t^j based on a linear motion assumption [36]:

$$\mathbb{A}_m(T_{t-1}^i, d_t^j) = \aleph(\tilde{p}^i, p_d^j), \tag{9}$$

where $\tilde{p}^i$ and p_d^j represent the positions of the target T_{t-1}^i and detection d_t^j, respectively, and $\aleph(.)$ is a Gaussian distribution function.

Then, an association score matrix $\mathbb{S}$ is used to express the affinity score between detections and tracks:

$$\mathbb{S} = \left[s_{ij}\right]_{n_h \times n_d}, s_{ij} = -ln(\mathbb{A}(T_{t-1}^i, d_t^j)). \tag{10}$$

The Hungarian algorithm [34] is used to determine the track-detection pairs with the lowest affinity value in $\mathbb{S}$. A detection d_t^j is associated with T_{t-1}^i when the association cost s_{ij} is less than a pre-defined threshold in [36].

IV. SYSTEM OVERVIEW

Our overall objective is to develop a framework capable of detecting and tracking different vehicles and record their trajectories from a video sequence in the Moroccan urban. To do this, we have two specific objectives to be achieved:

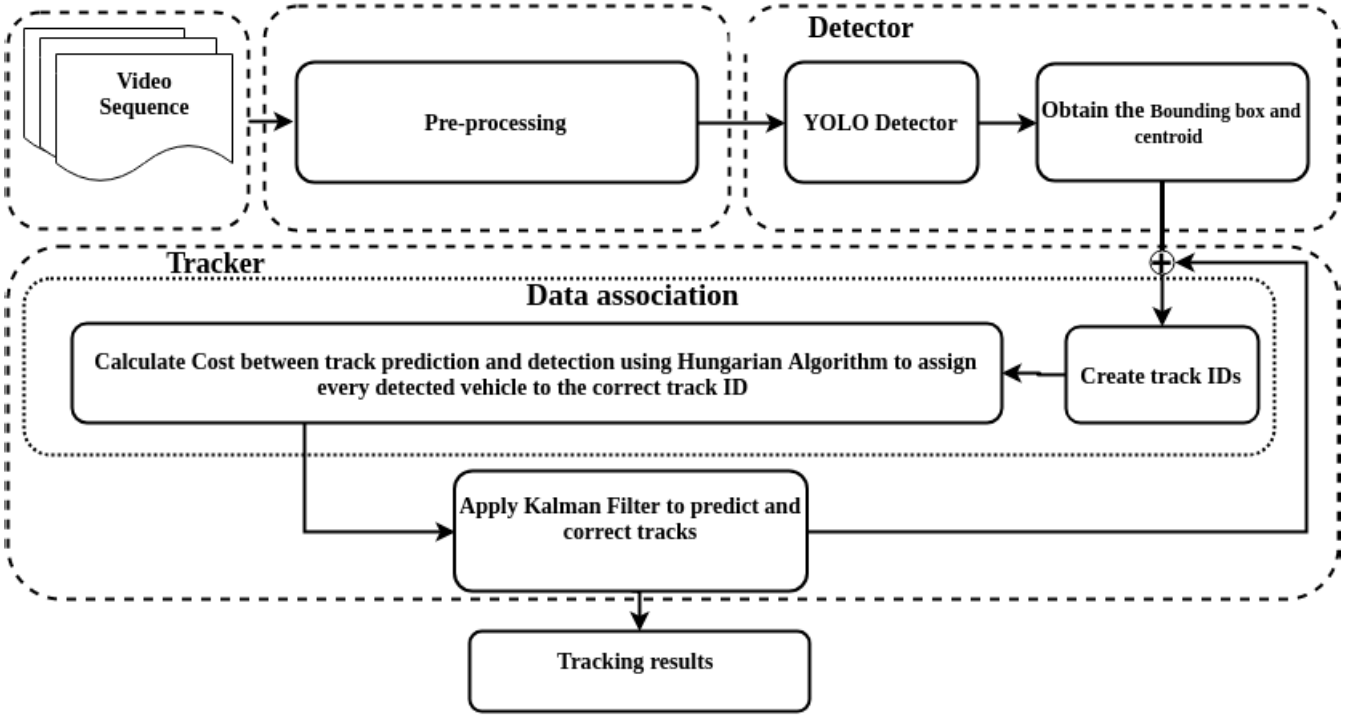


FIGURE 3. The MHDT framework modular design.

- Detect vehicles in the video sequence and resolve problems of detection in the urban environment, such as shadows, occlusion, change of illumination and weather conditions.
- Extract the trajectories of the vehicles by calculating the correspondence between the vehicles in the different frames via a strategy of data association and the elaboration of a tracking model.

To ensure good organization of the progress of the work, we used the benefits of modular design in our implemented approach. Figure 3 shows the modular design of the framework system.

The proposed approach MHDT for vehicle detection and tracking is composed of four blocks named Processing Block, Detector Block, Tracker Block, and Results Block. The functions of these blocks are as follows:

- **Pre-processing Block:** starts the video sequence and convert it into frames to be processed.
- **Detector Block:** detects the vehicles having the specified area (bounding box).
- **Tracker Block:** In this block we combine Kalman filter and data association-based tracking methods. Furthermore, we estimate the bounding box/labels (ID), the centroid, and the orientation of the vehicle tracker.
- **Results Block:** delivers the tracking trajectory of the vehicles on the basis of the region properties of the vehicles such as bounding box/labels (ID), and the centroid.

Algorithm 1 The Algorithm of the Proposed Framework MHDT Can Be Explained as Follows:

Input: Video sequence $V = \{I_i\}_{i=1}^{N}$, where N is the number of Images ;
Output: Trajectories T of the vehicles in the video ;
Initialization: $T \leftarrow \phi$;
foreach *Image I in V* **do**
    // Using YOLO detector to detects the vehicules;
    $D \leftarrow \{d_i\}_{i=1}^{N'}$ // N' The numbre of vehicles detected ;
    foreach *detected d in D* **do**
        // Create or update tracks and attribute track ID using Kalman filter and data association;
        $T \leftarrow T \cup \{d\}$;
    end
end

V. EXPERIMENTS RESULTS

A. DATASET

The context of this paper for detection and tracking of vehicles is the Morrocan Urban Network. In the best of our knowledge, there is no Morrocan traffic video dataset. Therefore, we called our dataset. MoVITS dataset (https://data.mendeley.com/datasets/5jcg5vfx58/3) [37] contains more than 75,230 images annotated with a bounding box (Ground Truth). It has been recorded under natural conditions using a stereo-vision system in a Moroccan urban area.

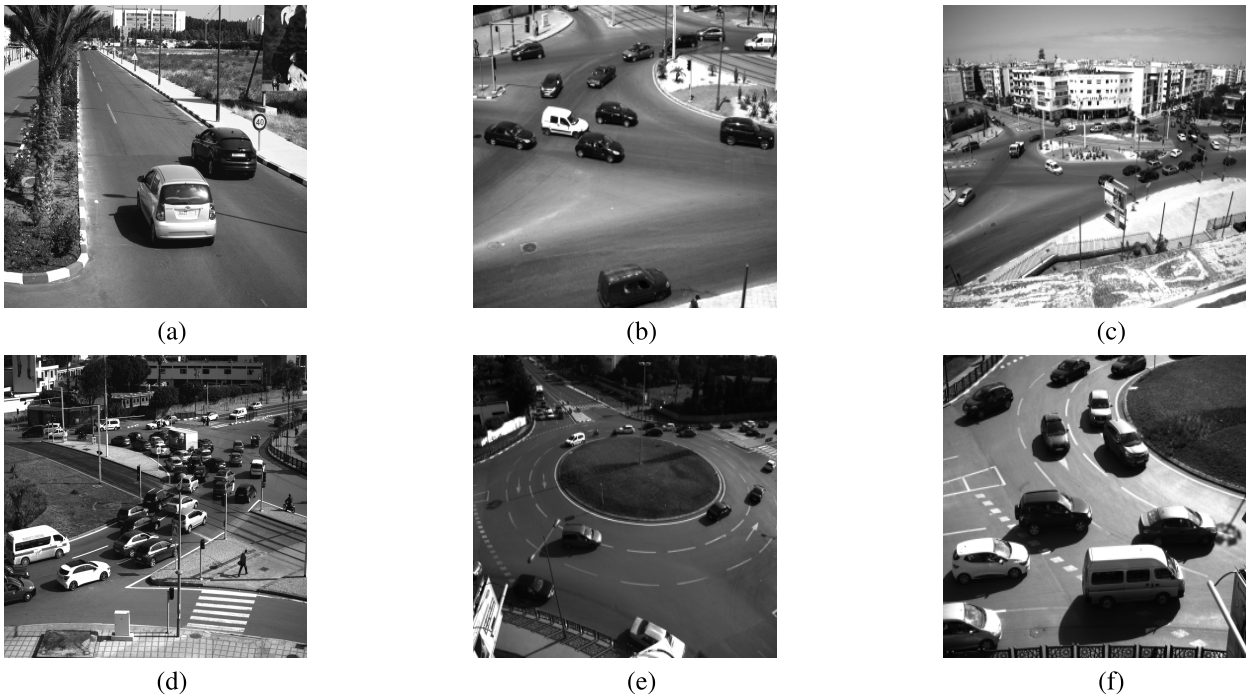


FIGURE 4. MoVITS dataset. (a) Highway, (b, c, d and f) Intersections and (e) Roundabout. [38].

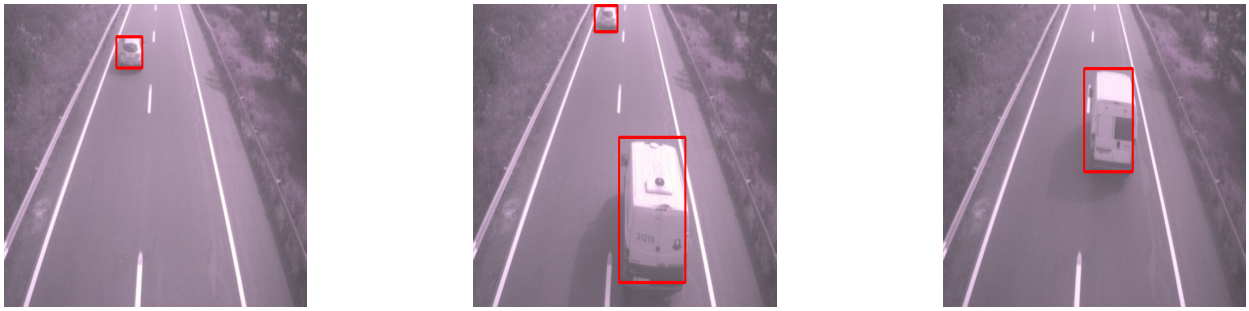


FIGURE 5. Detection results of the Highway sequence. Frames 1, 14 and 176 are displayed.

This Dataset contains challenging scenarios such as occlusion, shadows, varying illumination, and complex background. Figure 5 shows some example from the collected data. There are videos with 5 megapixels, 15 frames per second at a different time and places.

B. SETTINGS AND MATERIAL USED

We selected normalized RGB colour space as the feature space, and it was quantized into 16 × 16 × 16 bins for comparison between different algorithms. It should be noted that other colour spaces such as the HSV colour space can also be used in the MHDT framework. The MHDT component runs at 3.70GHz a single core of an Intel(R) Core (TM) i7-8700K machine with 16 GB memory, NVIDIA GeForce GTX 1080 Ti, Operating System Linux 64-bit. The program was implemented using C++ without any parallel programming, Qt Framework, and the Open Source Computer Vision (OpenCV) library.

C. RESULTS

The effectiveness of the proposed framework is evaluated using three videos (with a number of frames varying between a minimum of 1263 frames and a maximum of 15300 frames) from the MoVITS dataset. The experimental results show that the proposed MHDT framework achieves good estimation accuracy of the scale and the orientation of vehicles in the video sequences. Different sequences are used, and each sequence has its characteristics (rotation, pause, scale variation, and occlusion). We set up experiments to list the estimated width, height, and orientation of the vehicles.

We first use a Highway sequence (where the resolution is 2456 × 2054, the frame rate is 15 fps, and the number

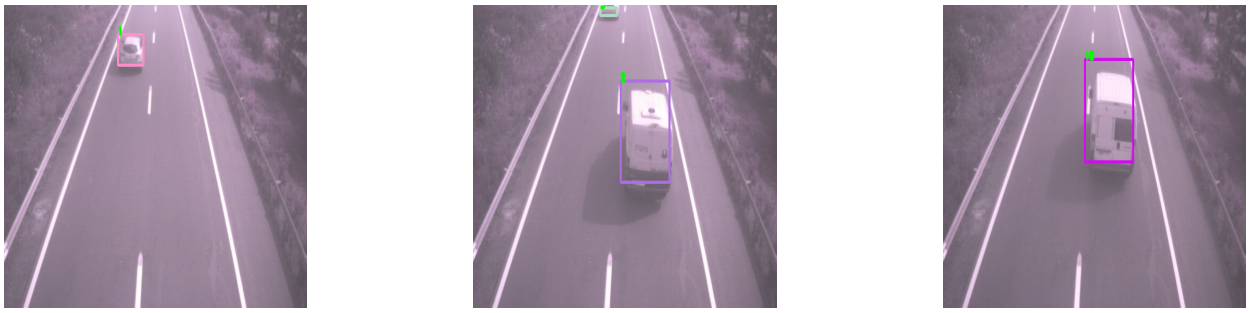


FIGURE 6. Tracking results of the Highway sequence. Frames 1, 14 and 176 are displayed.

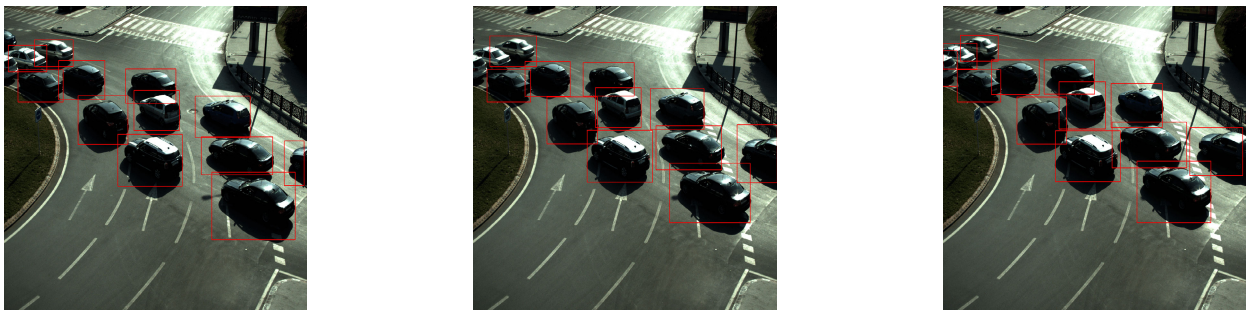


FIGURE 7. Detection results of the Intersection$_1$ sequence. Frames 700, 739 and 742 are displayed.

of frames is 6225) to verify the efficiency of the proposed approach. As shown in Figure 5 and 6, the external bounding box represents the target candidate regions used to estimate the real targets, that is, the inner bounding box. The experimental results show that the proposed Framework is robust when aiming at predicting the mean position and trajectory of the vehicles with scale and orientation changes.

The second test is an Intersection$_1$ sequence (where the resolution is 2456 $\times$ 2054, the frame rate is 15 fps, and the number of frames is 11,006), used to verify the efficiency of the proposed framework on a more complicated situation (Figure 7 and 8). The vehicles exhibit large scale changes with partial occlusion. However, the proposed system works much better in estimating the scale and orientation of the targets, especially when an occlusion occurs.

The last experiment is an Intersection$_2$ sequence (where the resolution is 2456 $\times$ 2054, the frame rate is 15 fps, and the number of frames is 10,321) presented in Figure 9 and 10. The experimental results show that the proposed framework estimates good accuracy of the scales and orientations of the targets, especially in the case of occlusions.

D. COMPUTATIONAL COMPLEXITY

In this section, we have provided a computational complexity of the MHDT. the time complexity for the remaining steps is summarized as follows:

For the detection step, the image is modified and altered to a size of 416 $\times$ 416 and then the image is put through a slice and dice system where they are divided into 7 $\times$ 7 size. This implies that the size of each grid is of size 64 $\times$ 64 and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an 64 $\times$ 64 $\times$ $(B * 5 + C)$ tensor.

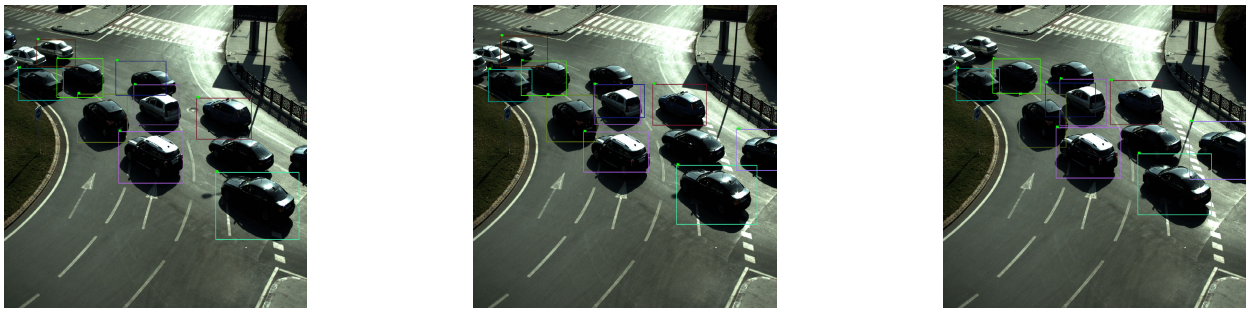
For the tracking step, the state prediction has a complexity of $O(b)$, and for state corrections it has a complexity of $O(b^2)$, and for data association $O(b^3)$, where b is the number of bounding boxes. The total computational cost is therefore $O(b^3 + b^2 + b)$.

E. PERFORMANCE MEASURES

The Precision is the ratio of positive-correctly-predicted observations to the total positive-predicted observations. The question that this metric answer is of all passengers that labelled as survived, how many survived. High precision relates to the low false positive rate. Precision is defined as the number of true positives T_p over the sum of the number of false positives and the number of true positives F_p:

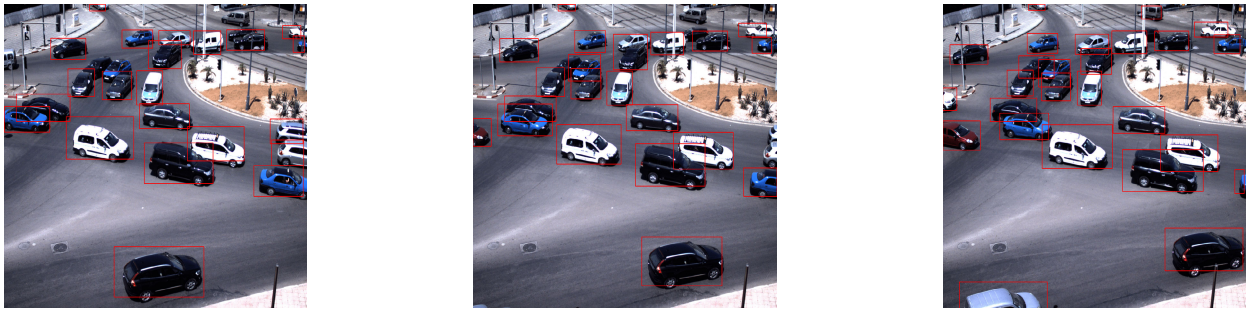
$$Precision = \frac{T_p}{T_p + F_p}. \tag{11}$$

The Recall is the ratio of correctly predicted positive observations to all observations in actual class. The recall indicates how many passengers from the ones that truly survived, were labelled as such. A recall is defined as the sum of the number



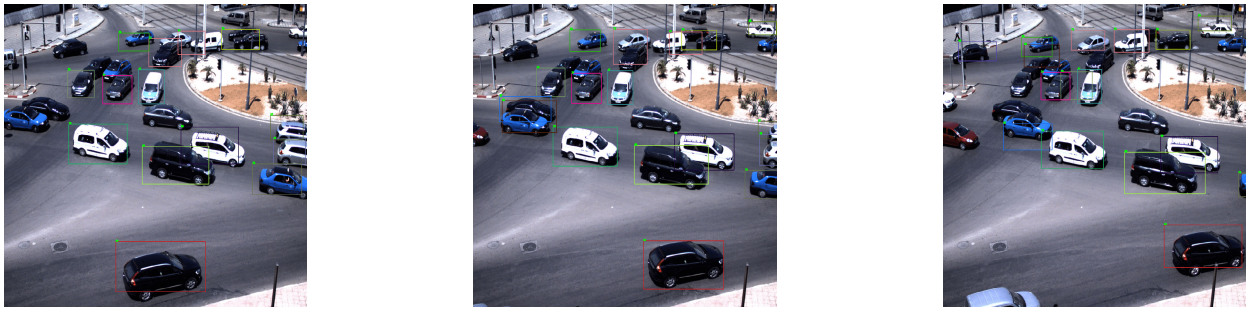
Frame 700 Frame 739 Frame 742

FIGURE 8. Tracking results of the Intersection$_1$ sequence. Frames 700, 739 and 742 are displayed.



Frame 200 Frame 210 Frame 240

FIGURE 9. Detection results of the Intersection$_2$ sequence. Frames 200, 210 and 240 are displayed.



Frame 200 Frame 210 Frame 240

FIGURE 10. Tracking results of the Intersection$_2$ sequence. Frames 200, 210 and 240 are displayed.

of true positives T_p over the sum of the number of false negatives and the number of true positives F_n:

$$Recall = \frac{T_p}{T_p + F_n}. \tag{12}$$

We studied the Precision-Recall curves of the proposed system. Figure 11 shows the performances of the proposed method in each video sequence used.

The F_1 Score is defined as the harmonic mean of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F_1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have a similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

The detection and Tracking performances are evaluated using the MoVITS dataset. Table 2 compares the proposed framework MHDT with two baseline trackers MDP [5] and SORT [6].

The experimental results demonstrate that the proposed MHDT is robust enough to detect and track the trajectory of the vehicles in different situations (scale variation, pause, rotation, and occlusion).

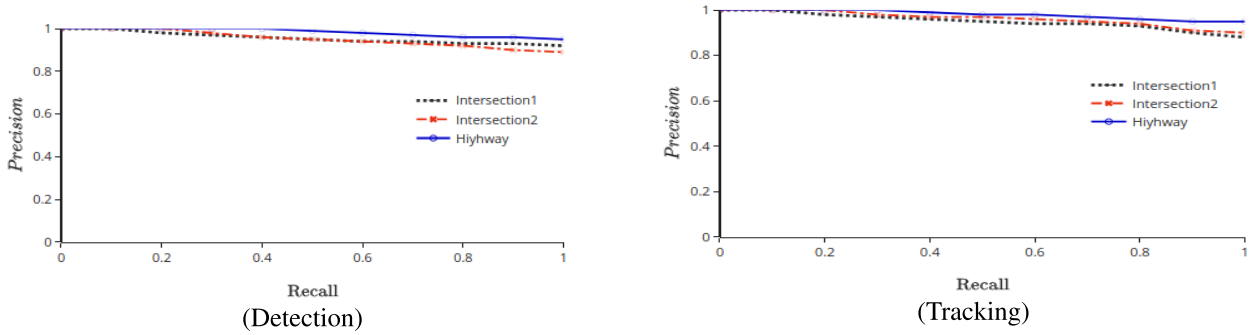


FIGURE 11. Precision-Recall curves of the proposed system.

TABLE 2. Performance of the proposed approach on the MoVITS dataset.

Methods	video Sequences	Detection			Tracking		
		Precision	Recall	F_1 Score	Precision	Recall	F_1 Score
MDP [5]	Intersection_2	56.7	63.4	59.86	60.1	65.8	62.82
	Intersection_1	60.4	70.3	64.97	63.9	70.5	67.03
	Highway	70.1	73.2	71.61	69.7	75	72.25
SORT [6]	Intersection_2	62.5	67.9	65.08	63.1	69	65.91
	Intersection_1	70	77.2	73.42	78	80	78.98
	Highway	88	85	86.47	83.6	88.5	85.98
MHDT (Proposed)	Intersection_2	96	94.1	95.04	92	93.2	92.50
	Intersection_1	97.2	96.3	96.74	93.3	94	93.64
	Highway	96	94.4	95.19	93.2	93.8	93.49

In the future, we plan to analyse the trajectory of the vehicles to manage Moroccan traffic and road violations. The aim is to establish an automatic management system to monitor traffic flow and detect road violation, to cope with the ascending raise in vehicles numbers and to reduce the accidents rates caused by the non-respect of traffic laws.

VI. CONCLUSION

In this paper, a novel Multiple Hypotheses Detection and Tracking approach have been presented for Moroccan traffic surveillance in complex scenes. In this approach, we combine a Kalman filter and data association-based tracking methods using the YOLO detection approach [9]. The newly MHDT has been compared with MDP [5] and SORT [6] algorithms using a MoVITS dataset collected in different condition, and different places. The experimental results demonstrate that the proposed approach is robust in detecting and tracking vehicles under different conditions (scale variation, pause, rotation, and occlusion). In the future research, we will focus on vehicle trajectory analysis in order to detect directional change violations, and take more control of traffic management and traffic violations.

REFERENCES

[1] B. Zhang and J. Zhang, "A traffic surveillance system for obtaining comprehensive information of the passing vehicles based on instance segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7040–7055, Nov. 2021.

[2] D. Srivastava, S. Shaikh, and P. Shah, "Automatic traffic surveillance system utilizing object detection and image processing," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2021, pp. 1–5.

[3] 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.

[4] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73340–73358, 2020.

[5] F. Wu, S. Peng, J. Zhou, Q. Liu, and X. Xie, "Object tracking via online multiple instance learning with reliable components," *Comput. Vis. Image Understand.*, vol. 172, pp. 25–36, Jul. 2018.

[6] J. Gwak, "Multi-object tracking through learning relational appearance features and motion patterns," *Comput. Vis. Image Understand.*, vol. 162, pp. 103–115, Sep. 2017.

[7] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2000, pp. 18–32.

[8] M. A. Naiel, M. O. Ahmad, M. N. S. Swamy, J. Lim, and M.-H. Yang, "Online multi-object tracking via robust collaborative model and sample selection," *Comput. Vis. Image Understand.*, vol. 154, pp. 94–107, Jan. 2017.

[9] 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.

[10] M. Han, W. Xu, H. Tao, and Y. Gong, "An algorithm for multiple object trajectory tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2004, p. 1.

[11] D. Riahi and G.-A. Bilodeau, "Online multi-object tracking by detection based on generative appearance models," *Comput. Vis. Image Understand.*, vol. 152, pp. 88–102, Nov. 2016.

[12] S. Huang, S. Jiang, and X. Zhu, "Multi-object tracking via discriminative appearance modeling," *Comput. Vis. Image Understand.*, vol. 153, pp. 77–87, Dec. 2016.

[13] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

[14] J. Prokaj, M. Duchaineau, and G. Medioni, "Inferring tracklets for multi-object tracking," in *Proc. CVPR WORKSHOPS*, Jun. 2011, pp. 37–44.

[15] J. Prokaj and G. Medioni, "Persistent tracking for wide area aerial surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1186–1193.

[16] A. Basharat, M. Turek, Y. Xu, C. Atkins, D. Stoup, K. Fieldhouse, P. Tunison, and A. Hoogs, "Real-time multi-target tracking at 210 megapixels/second in wide area motion imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 839–846.
[17] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
[18] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3542–3549.
[19] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.
[20] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, *arXiv:1608.07242*.
[21] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.
[22] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 472–488.
[23] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2711–2720.
[24] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3119–3127.
[25] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.
[26] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, "Robust object tracking based on temporal and spatial deep networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1153.
[27] C. Bregler and J. Malik, "Learning appearance based models: Mixtures of second moment experts," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 845–851.
[28] B. Han, J. Sim, and H. Adam, "BranchOut: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3356–3365.
[29] 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.
[30] A. A. Butt and R. T. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1846–1853.
[31] M. A. Abdelwahab, M. Abdel-Nasser, and M. Hori, "Reliable and rapid traffic congestion detection approach based on deep residual learning and motion trajectories," *IEEE Access*, vol. 8, pp. 182180–182192, 2020.
[32] M. A. Abdelwahab, M. Abdel-Nasser, and R.-I. Taniguchi, "Efficient and fast traffic congestion classification based on video dynamics and deep residual network," in *Proc. Int. Workshop Frontiers Comput. Vis.* Singapore: Springer, 2020, pp. 3–17.
[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME—J. Basic Eng.*, pp. 35–45, Mar. 1960.
[34] G. R. Waissi, "Network flows: Theory, algorithms, and applications," *Interfaces*, vol. 24, pp. 133–155, 1994.
[35] T. Jebara and R. Kondor, "Bhattacharyya and expected likelihood kernels," in *Learning Theory and Kernel Machines*. Berlin, Germany: Springer, 2003, pp. 57–71.
[36] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1218–1225.
[37] H. Derrouz, A. Elbouziady, H. A. Abdelali, R. O. H. Thami, S. E. Fkihi, and F. Bourzeix, "Moroccan video intelligent transport system: Vehicle type classification based on three-dimensional and two-dimensional features," *IEEE Access*, vol. 7, pp. 72528–72537, 2019.
[38] D. Hatim, A. A. Hamd, H. Rajae, M. F. Zahra, B. Omar, Z. Yahya, G. Mounir, O. H. T. Rachid, K. Ismail, B. Said, H. Abdelkrim, S. Nada, and B. Francois. *The Moroccan Video Intelligent Transport System Dataset for Vehicle Detection*. Accessed: 2021. [Online]. Available: https://data.mendeley.com/datasets/5jcg5vfx58/3

HAMD AIT ABDELALI defended his thesis at the Université Mohammed V de Rabat, in 2016. He is currently a Researcher at the Embedded System and IA Department, MAScIR, Rabat, Morocco.

HATIM DERROUZ (Member, IEEE) is currently pursuing the Ph.D. degree with the ENSIAS, Université Mohammed V de Rabat, Rabat, Morocco. He is currently a Postdoctoral Researcher at the Embedded System and IA Department, MAScIR, Rabat.

YAHYA ZENNAYI is currently the Head of the imaging pole at the Embedded Systems and AI Department, Moroccan Foundation for Advanced Science, Innovation and Research.

RACHID OULAD HAJ THAMI is currently a Full Professor of computer engineering at the ENSIAS, Rabat IT Center, Université Mohammed V de Rabat, Rabat, Morocco.

FRANÇOIS BOURZEIX is currently the Director of the Embedded Systems and AI Department, Moroccan Foundation for Advanced Science, Innovation and Research.