Future Generation Computer Systems 113 (2020) 506–527

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A novel multi-view pedestrian detection database for collaborative Intelligent Transportation Systems

Anouar Ben Khalifa [a,*], Ihsen Alouani [b], Mohamed Ali Mahjoub [a], Atika Rivenq [b]

[a] Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, LATIS- Laboratory of Advanced Technology and Intelligent Systems, 4023, Sousse, Tunisie

[b] IEMN-DOAE, Université polytechnique Hauts-de-France, Valenciennes, France

ARTICLE INFO

Article history:
Received 4 January 2020
Received in revised form 28 May 2020
Accepted 12 July 2020
Available online 16 July 2020

Keywords:
Multi-view
Environment perception
Collaborative intelligence
Pedestrian detection
Infrastructure to vehicle
CNN

ABSTRACT

Recent advances in machine-learning, especially in deep neural networks have significantly accelerated the development and deployment of transport-oriented intelligent designs with increasingly high efficiency. While these technologies are exceptionally promising toward revolutionizing our current mobility and reducing the number of road accidents, the way to safe Intelligent Transportation Systems (ITS) remains long. Since pedestrians are the most vulnerable road users, designing accurate pedestrian detection methods is a priority task. However, traditional monocular pedestrian detection methods are limited, especially in occlusion handling. Hence, a collaborative perception scheme in which vehicles no longer restrict their input data to their immediate embedded sensors and rather exploit data from remote sensors is necessary to achieve a more comprehensive environment perception. In this work, we propose a novel public dataset: Infrastructure to Vehicle Multi-View Pedestrian Detection Database (I2V-MVPD) that combines synchronized images from both a mobile camera embedded in a car and a static camera in the road infrastructure. We also propose a new multi-view pedestrian detection framework based on collaborative intelligence between vehicles and infrastructure. Our results show a significant improvement in detection performance over monocular detection.

1. Introduction

Nowadays transportation systems are undergoing disruptive transformations. With various technologies that are revolutionizing the sector such as connected cars and artificial intelligence-driven technologies, the automotive ecosystem is shifting toward new design paradigms. Recent advances in machine learning, especially in deep neural networks have significantly accelerated the development and deployment of automotive-oriented intelligent designs with increasingly high efficiency [1–3]. While these technologies are exceptionally promising toward improving our current mobility habits and reducing the number of road accidents, serious challenges to ITS-driven safe mobility remain to solve [4,5]. In fact, while in-vehicle embedded intelligence brought impressive results especially for environment perception, these systems still remain unable to overcome complex misleading traffic situations [6–8]. A real time collaborative perception scheme in which cars no longer restrict their input data to their immediate embedded sensors and exploit data from remote sensors is necessary for a reliable environment perception [9,10]. In fact, vehicles are no longer isolated systems controlled only by their drivers, but rather assimilated to distributed intelligent nodes within an interconnected complex system. With vehicular wireless communication, ITS applications are supported by several high-performance communication technologies such as ITS-G5 and 5G [6,11]. Consequently, vehicles are able to interact actively and in a participative manner using On Board Units (OBUs) and the infrastructure's Road Side Units (RSUs). Several applications that use vehicular communication have been proposed in the context of ITS, such as intersection safety [10,12], traffic lights and signs detection [13], safe driving [14,15], visibility estimation [16,17], road monitoring [5,18,19], etc. Other applications tackle the problem of pedestrian detection which is a cornerstone of any autonomous vehicle [20–22]. However, none of these works make use of a collaborative intelligence i.e. intelligent systems communicating in a collaborative manner with the infrastructure.

In the case of pedestrian detection applications, existing works are mainly based on monocular detection systems (using only one camera). While these systems have obtained good results especially with the development of Convolutional Neural Networks (CNN) [1,3], they remain limited by challenging problems such as occlusions [7,23]. These limitations motivated the development of multi-view pedestrian detection systems, which have been able to overcome monocular perception limitations. The use of

* Correspondence to: ENISo, 4023, Sousse, Tunisia.
E-mail address: anouar.benkhalifa@eniso.rnu.tn (A. Ben Khalifa).

https://doi.org/10.1016/j.future.2020.07.025
0167-739X/

multiple cameras introduces its own challenges such as synchronizing and combining the images [24,25]. These techniques were made possible due to public multi-view datasets such as [26–30]. However, none of these databases considers static cameras in the infrastructure combined with a mobile in-vehicle camera.

Motivated by these challenges, we address a problem with a significant impact on transportation systems safety. In this article, we tackle the specific problem of multi-view pedestrian detection in a vehicular environment. We introduce the first real-world dataset named Infrastructure to Vehicle Multi-View Pedestrian Detection Dataset (I2V-MVPD). To the best of our knowledge, this is the first multi-view pedestrian detection database available to the scientific community combining both a static camera mounted on a road infrastructure and a mobile camera embedded in a vehicle's dashboard. This work encourages the community to develop techniques for collaborative intelligence by offering an open dataset[1] for research and benchmarking. Based on this dataset, we propose a novel framework for road users' detection using a collaborative scheme through Infrastructure to Vehicle (I2V) communication. In our framework, the first step is to prepare the monocular detector based on deep transfer learning scheme using training data from both views. Then, we deploy our trained detector on each view separately to obtain monocular detection results. To compensate the camera non-calibration, we identify correspondence points between the two views and calculate geometric transformation parameters. Finally, we use these projections to refine the vehicle detection results.

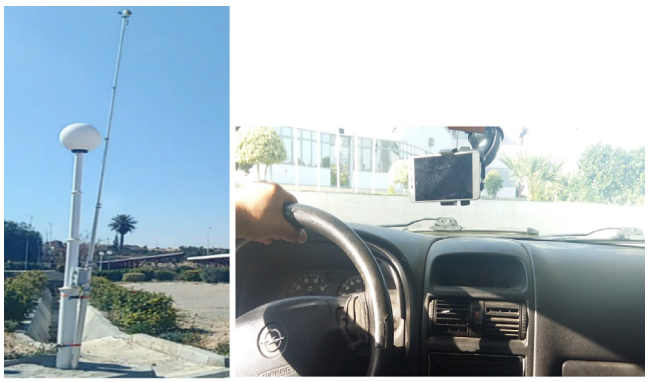
The main contributions of this paper are as follows:

- We introduce a novel public Infrastructure to Vehicle Multi-View Pedestrian Detection dataset that is the first to combine synchronized images from both a mobile camera installed in a vehicle and a static camera installed on the roadside.
- We propose a new Multi-View pedestrian detection framework for uncalibrated cameras, we demonstrate that this cooperative perception improves the performance of pedestrian detection systems.
- We propose an exhaustive literature review of V2I communication-based computer vision techniques.
- We give a thorough overview on available multi-view datasets for pedestrian detection.

The remainder of the paper is organized as follows. In Section 2, we introduce an overview of I2V and V2I communication-based computer vision techniques. In Section 3, we present the currently available multi-view pedestrian detection databases. In Section 4, we summarize related work on multi-view pedestrian detection, showcase their results and explore methods to overcome the lack of calibration data. In Section 5, we present our Infrastructure to Vehicle Multi-View Pedestrian Detection dataset. In Section 6, we present our collaborative deep multi-view detection framework. The experimental results and analysis are shown in Section 7. Finally, we conclude the paper with a summary and future research in Section 8.

2. Vehicular communication-based environment perception systems

Connected and autonomous vehicles are becoming the center of attention of transportation-oriented researchers in both academia and industry. This is confirmed by the increasing number of projects led by companies such as Google with its self-driving car taxi service project[2] and Tesla which commercializes



(a) (b)

Fig. 1. The camera setup: (a) Static, (b) Mobile.

vehicles equipped with autopilot mode and working toward a full self-driving system[3] [9]. To reach a reliable environment perception and thereby ensure safety and user comfort, intelligent transportation systems need to operate in collaborative manner with each other and with smart infrastructure elements [4]. This is possible due to Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Infrastructure to Vehicle (I2V) communication technologies. In this work, we focus on computer vision-based V2I and I2V collaborative setting to improve the safety of the intelligent transport systems. Several problems have been tackled based on V2I communication such as intersection safety [10,12], traffic lights and signs detection [13], pedestrian and cyclist detection [20–22], visibility estimation [16,17], road monitoring [5,18,19], etc.

To make road intersections safer, Vishnu et al. [12] propose an infrastructure-side module equipped with a camera, a processing unit and I2V communication module. The camera detects moving objects and their directions using the Lucas–Kanade optical flow algorithm and sends a warning to nearby vehicles with the I2V module. Goldhammer et al. [10] propose to equip an intersection with an array of 10 monochrome cameras and 14 laser scanners placed high above the ground to provide a bird's eye view and collect distance and texture information. A processing unit uses this data to determine the locations of pedestrians and cyclists. The locations are then sent to a roadside communication unit that broadcasts these locations to nearby vehicles using wireless I2V communication.

To read the state of traffic lights and signs, García-Garrido et al. [13] proposed a system to recognize traffic signs using a pair of cameras, with an I2V communication system to filter road signs irrelevant to the car's current status. To detect the road signs, the authors first transform the color image into grayscale since it is more robust against bad lighting conditions. Then a Canny edge detector is applied to the image, and a Hough transform is applied to the resulting contours in order to detect triangular or circular signs. A Support Vector Machine (SVM) classifier is then used to recognize the obtained signs among a selection of 100 signs. An I2V system where transmitters are placed on traffic signs and receivers are installed on vehicles is used to transmit information such as the road name, traffic direction and GPS localization to determine whether the sign is of interest to the vehicle's current path.

To detect pedestrians, Al-Refai et al. [22] introduced a system that detects pedestrians in images taken from nearby vehicles.

[1] https://sites.google.com/site/benkhalifaanouar1/6-datasets.

[2] https://waymo.com/

[3] https://www.tesla.com/autopilot.

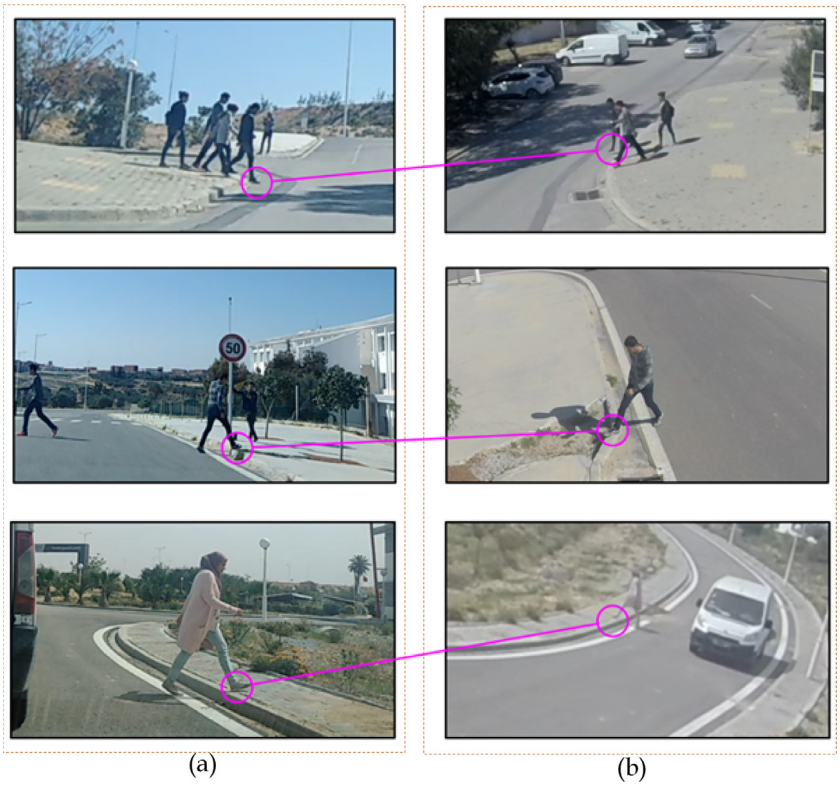


Fig. 2. Illustration of the synchronization precision. (a) Vehicle view, (b) infrastructure view.

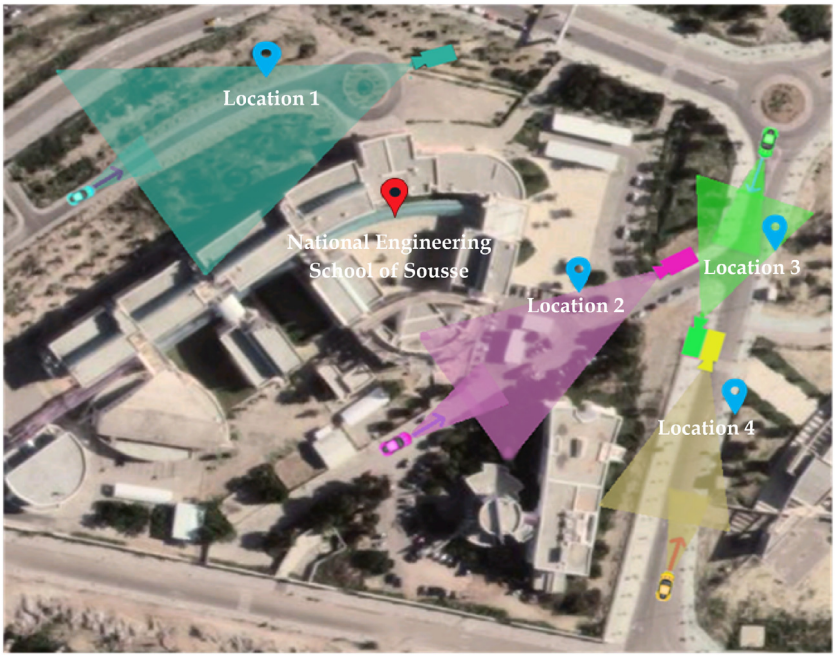


Fig. 3. Plan view showing the acquisition locations.

First, the vehicle sends video frames and GPS data to the infrastructure processing unit using V2I communications. Then, the infrastructure may update its internal database with the new images after pre-processing them. The database images are used to create a background model for each GPS location using the Gaussian Mixture Model method. Finally, moving objects are detected for object proposals by comparing the frame to the background, and pedestrians are detected using HOG features and an SVM classifier. Zhao et al. [21] proposed another pedestrian detection approach based on LIDAR sensors and an artificial neural network. The LIDAR point cloud is filtered and then clustered to detect and track pedestrians and vehicles. Traffic lanes are also detected by aggregating vehicle paths. This data is then used to obtain road user trajectories. A neural network is then trained to use this trajectory data and predict whether a pedestrian will cross the road or not. The authors assert that this method can be used

in conjunction with vehicular communications in order to warn road users of potential pedestrian crossings.

To estimate visibility in foggy environments, Hautiére et al. [16] propose a data fusion framework. A roadside unit collects data from its sensors and nearby vehicles' sensors using V2I communications to calculate local visibility estimates. This estimate can then be used to issue warnings or speed recommendations to vehicles. Chaabani et al. [17] proposed another framework that estimates visibility ranges based on global features and a neural network classifier. To detect the density of the fog, the authors use a single camera mounted on a car or on the roadside. Global features are extracted from the images using a Fourier transform and the distribution of gray levels. A three-layer neural network is trained to classify the fog levels in 6 bins using the obtained features as an input. The authors conclude that their system can be implemented on existing infrastructure which can issue warnings or speed recommendations to nearby drivers.

To monitor the state of the road and identify potential risks, Taie and Taha [18] present a system to monitor traffic and generate periodic reports on any events that may happen. This system is composed from road-side units (RSUs) and OBUs with cameras on the vehicles. The vehicles can send video sequences to the infrastructure which analyzes them to detect 3 types of events (accident, emergency vehicle passing, crowds of people) by applying a Multiple Gaussian Model and a SURF detector. A report is then generated and sent to a control center and nearby vehicles that can relay it to other vehicles via V2V communication. Zhao et al. [19] propose a method to monitor the local urban driving environment by fusing data from a camera installed on the infrastructure side and GPS data from nearby vehicles. The system uses an RSU with a camera that first detects the different road lanes by extracting lane markers. The RSU then detects and tracks vehicle locations by using SIFT features. Meanwhile, the vehicles send their GPS locations to the RSU which uses both sets of data to create a road occupancy map. Jiménez et al. [5] propose a system to detect and identify obstacles in rural or intercity roads by using an onboard LIDAR and a stereo camera as well as V2I and V2V communication systems. HOG features are extracted from the stereo camera and the LIDAR point cloud is reconstructed and aligned with the camera view according to a new clustering method proposed in the paper. Objects from both sensors are then classified with an SVM approach. Vehicles can then communicate obstacle positions to other vehicles and to the infrastructure.

To improve general road safety, Olaverri-Monreal et al. [4] implemented a system to check if the car behind the driver is respecting the safe driving distance and warn them if not. This system is composed of a stereo camera, a processing unit, a warning display and VANET communication equipment. The following vehicle is detected in both views using a Viola Jones Haar Cascade detector, and the distance between the two cars is calculated. If this distance is less than the safe distance, a message is shown in the back of the car, and a warning message is sent to the following car by via V2V communication. In case of the sudden appearance of a potential danger, Hirano et al. [31] propose a framework to assist in evasive maneuvers by using a network of high-speed cameras that communicate between them. The authors opt for a background subtraction method to locate the targets and obtain their positions. An I2V network is used to communicate the target positions to the vehicle which can then decide whether or not to engage in evasive maneuvers based on this data. The main advantage of this method is its low latency, which is necessary in order to avoid danger in time.

An analysis of the different works that combine I2V/V2I communications and computer vision shows that in most cases, only the infrastructure side or the vehicle side possesses an intelligent processing unit, the other parties only communicate data or receive decisions. To the best of our knowledge, there are no works that make use of collaborative intelligence in which both the infrastructure and the vehicle actively participate in processing the information and taking the necessary decisions. This may be due to the lack of datasets that provide relevant data.

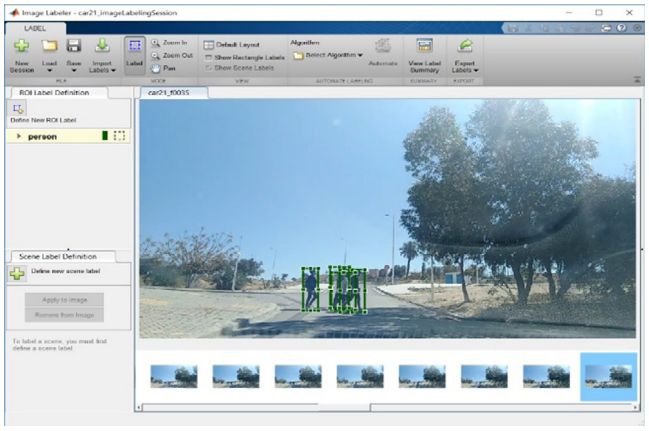


Fig. 4. The MATLAB image labeler application with annotation samples.

Fig. 5. Examples of bad illumination in our database. Person hidden in the shadow, Glare from reflected sunlight, Lighting differences caused by weather.

3. Multi-view pedestrian detection databases

To simulate real world conditions, many databases have been proposed, which we can divide in two categories: databases that use static cameras and databases that use mobile cameras. In this study, we focus on multi-view databases, and we define multi-view as the presence of images from two or more cameras.

3.1. Static multi-view pedestrian detection

In this section, we present seven static multi-view pedestrian detection databases. These databases use multiple fixed cameras in varied environments (indoor or outdoor) to capture the images. While some of these databases have different goals (such as people detection, crowd counting, group tracking…), they are still useful for pedestrian detection.

Fleuret et al. [26] proposed the EPFL database: 6 sequences (4 outdoors and 2 indoors) filmed with 4 cameras at 320×240 resolution and 25 fps. The low resolution of the cameras used in this dataset leads to poor image quality, and the presence of lighting changes, occlusions and shadows further increase the challenges found in this dataset. Ferryman and Shahrokni [27] proposed the widely used PETS 2009 dataset for crowd counting and pedestrian tracking. This dataset contains multiple sequences filmed with up to 8 cameras at a 720×576 resolution and around

Table 1
Overview of static multi-view pedestrian detection databases.

Ref.	Databases	Year	Nb. Camera	fps	Resolution	Size	Challenges			
							Occlusions	Lighting changes	shadows	Complex background
[26]	EPFL	2008	4	25	320×240	∼15K frames	✔	✔	✔	
[27]	PETS 2009	2009	8	7	720×576	795 frames	✔	✔	✔	
[35]	SALSA	2015	4	15	1024×768	60 min.	✔	✔		
[28]	Campus	2016	4	30	1920×1080	4×4 min.	✔	✔	✔	✔
[30]	DukeMTMC	2016	8	60	1920×1080	85 min.	✔	✔		
[29]	EPFL-RLC	2017	3	60	1920×1080	8K frames	✔			
[32]	WILDTRACK	2018	7	60	1920×1080	60 min.	✔	✔	✔	✔

Table 2
Overview of mobile multi-view pedestrian detection databases.

Ref.	Databases	Year	Nb. Camera	fps	Resolution	Size	Challenges				
							Occlusions	Lightning changes	Shadows	Complex background	Camera problems
[34]	ETH	2008	2	15	640×480	4203 frames	✔	✔		✔	✔
[36]	Daimler Stereo	2011	2[a]	15	640×480	∼70K frames		✔		✔	
[37]	KITTI	2012	4[b]	10	1392×512	7 min	✔	✔	✔	✔	
[38]	KAIST	2015	2[c]	20	640×480 (RGB) 320×256 (IR)	∼95K frames	✔	✔		✔	✔
[39]	CVC-14	2016	2[d]	20	1280×1024(RGB) 640×512 (IR)	∼14K frames	✔	✔		✔	

[a]Grayscale stereo camera.
[b]Two RGB cameras and two infrared cameras.
[c]One RGB camera and one infrared camera.
[d]One grayscale camera and one infrared camera.

7 fps. The most used sequence for pedestrian detection is the S2-L1 sequence which is 795 frames long [28,32–34]. This sequence contains multiple challenges to overcome: occlusions due to people walking next to each other or due to object, lighting variation between the different views and the presence of shadows. Alameda et al. [35] proposed the SALSA dataset in order to study conversational groups in social settings. One part of this study involves detecting and tracking people. This dataset provides over 60 min of video taken from four different cameras in two different indoor environments (along with other sensors). The videos were taken at a resolution of 1024×768 and 15 fps. The video suffers from illumination variations and heavy occlusions due to the crowded scenes. Xu et al. [28] have proposed the Campus dataset. It is composed of four sequences of four minutes each. Every sequence has been filmed with 4 HD cameras at 30 fps. Three sequences have been filmed outdoor and a sequence has been filmed indoor. The outdoor scenes have complex backgrounds (trees, poles, buildings) and other objects in the foreground that cause frequent occlusions. Shadows can also cause further challenges in this dataset. In [30], Ristani et al. proposed the DukeMTMC dataset for tracking multiple targets across multiple cameras. This database consists of 85 min of 60 fps high definition video filmed with 8 static cameras. While there are some blind spots in the camera configuration, enough overlaps exist in this database that can be used for multi-view pedestrian detection. The main challenge in this dataset is occlusion because of the large number of people walking. To test their people detection method, Chavdarova and Fleuret [29] created a new public dataset, EPFL-RLC. This dataset consists of 8000 frames taken from three static HD cameras filmed in the interior of a building. Objects in the area and people walking in groups cause some occlusions. Recently, Chavdarova et al. [32] proposed the WILDTRACK dataset, which has 60 min of 60 fps footage captured from 7 highly overlapping static HD cameras (four GoPro Hero3 and three GoPro Hero4) in front of a university's main building. This large-scale dataset features many challenges to overcome: complex backgrounds, large crowds that cause a lot of occlusions and the presence of shadows make pedestrian detection a difficult task. Table 1 gives a quick overview on the static databases state-of-the-art.

3.2. Mobile multi-view pedestrian detection

In this section, we present five mobile multi-view pedestrian detection databases. The authors collect these databases by installing cameras on mobile platforms. In most of the cases, the mobile platform is a car and the images are taken in an urban environment. This follows the trends of increased interest in safety for autonomous and assisted driving.

Ess et al. [34] proposed the ETH dataset, which was filmed from two cameras installed on top of a moving platform. The dataset contains 4203 frames extracted from 5 sequences at 15 fps and 640×480 resolution. The sequences were acquired from walking with the platform in an urban setting. This creates many challenges: camera ego-motion, complex backgrounds, occlusions and lighting variations. As an extension of their existing monocular pedestrian detection dataset, Keller et al. [36] proposed the Daimler stereo dataset. A stereo camera is installed on top of a car that is driving in an urban environment. The frames have a resolution of 640×480 and are taken at 15 fps. The images are in grayscale and their low resolution compounded with the complex backgrounds constitute the difficulties encountered in this dataset. For the purpose of autonomous driving research, Geiger et al. [37] proposed the KITTI dataset. This dataset uses a stereo camera and 2 grayscale cameras (along with LIDAR and GPS) to acquire the images. These cameras and instruments were installed on a car and real-world data was acquired by driving in an urban setting. This results in a cluttered environment and complex backgrounds. Occlusions and shadows are also a further source of difficulty in this dataset. Hwang et al. [38] proposed the KAIST database, which is the first pedestrian dataset that combines both a visible light camera and an infrared camera. The dataset contains 95 328 pairs of images acquired from a setup that uses a beam splitter and the two cameras. This setup was installed on the roof of a car and the images were acquired while driving in an urban environment at day and at night. The low resolution of the images, the complex backgrounds and the occlusions are the biggest difficulties encountered in this dataset. To showcase the potential improvement of infrared image acquisition and to provide a unified benchmark, Gonzalez et al. [39]

Table 3
Overview of multi-view pedestrian detection methods.

Ref	Methods	Databases	Performances
[26]	Monocular detection using CNN based background subtraction. Fusion using probabilistic occupation maps.	WILDTRACK	MODA = 0.232 MODP = 0.305 Precision = 0.75 Recall = 0.55
[40]	Feature extraction using a CNN and a Conditional Random Field will be used to create a probabilistic occupation map which will detect the pedestrians.	WILDTRACK	Precision = 0.95 Recall = 0.80
		EPFL Terrace	Precision = 0.88 Recall = 0.82
		PETS 2009 S2 L1	Precision = 0.93 Recall = 0.87
[28]	Monocular detection using Faster R-CNN. The detection fusion is made by checking 3D proximity.	WILDTRACK	MODA = 0.113 MODP = 0.184
		PETS 2009 S2 L1	MODA = 0.9 MODP = 0.74
		EPFL Terrace	MODA = 0.72 MODP = 0.71
[41]	Halfway fusion (after layer 4) of 2 Faster R-CNNs, pre-trained using the VGG-16 architecture.	KAIST	Miss Rate = 36.99%
[42]	Unsupervised detection using a region proposal network formed by the fusion of 2 VGGNet CNNs.	KAIST	Miss Rate = 36.42%
[43]	Fusion of 2 region proposal networks for Region of Interest detection, classification is assured by Boosted Decision Trees.	KAIST	Miss Rate = 29.83%
[44]	For two aligned visible and thermal images, Illumination Fully CNN computes the illumination-aware weights to calculate whether it is daytime frame or nighttime one. Through this illumination-aware mechanism, Illumination-Aware Two-Stream deep CCN make use of multi sub-networks to generate detection results and segmentation masks.	KAIST	Miss Rate = 26.37%
[45]	Multispectral pedestrian detection with different multispectral feature fusion strategies (concatenation, maximum, and sum). Deep Neural Network architecture for feature extraction and semantic segmentation.	KAIST	Miss Rate = 26.67%
[46]	Multi-layer fused CNN for multispectral pedestrian detection. A region proposal stage to combine the visible and thermal information. A detection stage to extract features from three feature maps and all features are combined through the fused ROI pooling layer.	KAIST	Miss Rate = 25.65%
[47]	A deformable part model is used in monocular candidate selection, these candidates will be matched to their corresponding candidates in the other view, and in the next and previous frames to confirm the detection.	ETHZ	Miss Rate = 43.66%
		KITTI	Miss Rate = 62%
		Daimler Stereo	Miss Rate = 80%

proposed the CVC-14 dataset. This dataset combines images from a visible light and an infrared camera. The two cameras were installed on a car and the sequence was acquired while driving around in the city. The 8518 pairs of images are acquired at a resolution of 1280×1024 for the visible light camera and 640×512 for the IR camera. The visible light camera suffers from lighting variations and low quality at night, combined with the low resolution of the IR camera forms the challenges of this dataset. Table 2 gives a quick overview of the mobile databases.

Interest in multi-view pedestrian databases has increased in the last few years, and this can be attributed to two main factors: the large number of applications that can be developed using these databases, and the desire to bypass the limits of single-view perception. The current trend in multi-view camera databases are to increase the size and the number of cameras in the database, while also increasing the quality and the frame rate [48,49]. The increase in database sizes is motivated by the recent development in deep learning techniques which require high amounts of data for training. Another trend specific to mobile databases is the use of multi-spectral imagery and other embedded sensors. The most common challenges encountered in these databases are occlusions which frequently happen due to groups of people walking together or due to foreground objects, illumination variations between views caused by the varying angles and positions of the cameras and complex backgrounds due to the urban environments.

Notice that none of these datasets consider infrastructure/vehicle multi-view or any hybrid database that combines static and mobile cameras.

4. Related works on multi-view pedestrian detection

Monocular pedestrian detection has always been a subject of great interest, with a rich literature on the subject. With the recent advances in CNN based methods, the precision and performance of monocular pedestrian detection is considered reliable in many scenarios [32,33]. However, these methods still have difficulties solving certain challenges such as pedestrian appearance changes, non-rigid deformations and occlusion. Further details on these problems can be found in [7,47,50]. These challenges led to an increased interest in multi-view pedestrian detection, since the blind spots of one camera can be recovered by other cameras [51]. In this section, we present an overview of existing multi-view pedestrian detection approaches. These approaches can be divided in two categories. The first category requires the

Fig. 6. Examples of appearance variation in our database.

Fig. 7. Examples of occlusions in our database.

Fig. 8. Example of background elements that cause false positives in our database.

use of calibrated cameras, while the second combines images from uncalibrated cameras.

Among the approaches that use calibrated cameras, Zhang and Tao [47] proposed a method for pedestrian detection in a binocular stereo setting that leverages information from both left and right views in a three-frame sequence to improve detection rates. A monocular deformable part model detector is applied to both views and a score is attributed to each detection candidate. The next step is to match detection candidates on the left frames with their corresponding detection in the right frame and in the next and previous frames. Each candidate is divided into overlapping patches and SIFT features and Lab color histograms are extracted from each patch. A similarity score is then calculated and the candidates with the highest similarity scores are matched. Park et al. [52] proposed a deep learning based multispectral pedestrian detection method that relies on images from calibrated RGB and infrared cameras. The first step is extracting features from the visible light and infrared images using a VGGNet architecture and a third fusion feature channel is then created by concatenating the RGB and IR feature channels. The next step consists of using the fusion features channel to detect candidates using a Region Proposal Network. The final step is an inference network used to determine the final detections from the region proposals. Chavdarova et al. [29] presented an end-to-end architecture based on the fusion of multiple CNNs. The CNNs are fine-tuned on the Caltech database. To improve robustness to occlusions, training data is augmented with images after applying an occlusion mask to them. To learn how multi-view features interact with each other, a multi-layer perceptron classifier that takes the output of the previously created CNN architecture as input is created and trained. The detection candidates obtained are projected on the ground plane, and non-maximum suppression is applied to select the strongest detection. Recently, López-Cifuentes et al. [33] proposed a method with promising results. First, a monocular pedestrian detector based on Faster R-CNN is applied to each view and the images are segmented using a CNN-based method. The segmentation results are projected on the ground plane to create an area of interest (AOI) in which a pedestrian could be present. The next step is to project the monocular detection on the ground plane and any detection outside of the AOI is discarded. To fuse the monocular detections, a geometrical approach is adopted: detections that are close to each other and have different source cameras are combined as a connected component of a graph. Finally, the arithmetic mean of each component is calculated and back-projected on the different views. Table 3 gives an overview on multi-view pedestrian detection methods as well as their performances on multiple databases.

Analyzing the previous methods shows that calibrated cameras-based approaches fuse the image data by projecting each view to a ground plane and projecting the results back to each view. However, if the cameras are uncalibrated, this fusion becomes very difficult, since it is not possible to directly obtain precise location data. In this case, many authors propose methods to combine information from different views in an uncalibrated camera network. Kang et al. [24] propose an approach to fuse multi-view information in an uncalibrated network for a pedestrian tracking application by creating a homography from a set of four matching points to register each camera to the ground plane. A spatio-temporal homography is then created using the previously calculated homography. A subsequent new homography is calculated through four points from the track and by calculating any alignment errors. This is used to keep tracking persons after leaving a camera's field to another camera's field. Varga et al. [25] propose a way to track people across a network of uncalibrated color and infrared cameras. First, lens distortions in the camera are corrected. Then the views are matched by calculating co-motion statistics. Coordinates where motion happens are detected between successive frames in each view. Next, to find correspondence points between each pair of views, concurrently changing pixels are detected and their similarity is measured by comparing their history of detected changes. Points that are not within the common field of view of both cameras are discarded. Finally, using the correspondence points, the images are aligned to create a ground plane. To solve this problem in the context of video summarization, Panda and Roy-Chowdhury [53] propose an approach without assuming prior knowledge about the correspondence between the views, which is the case if the cameras are uncalibrated. First, each video is split into multiple non-uniform shots, and a feature vector is extracted from each shot. These feature vectors are then used to calculate sparse coefficient matrices that represent intra-view or inter-view similarities between shots. Next, the rows of this matrix are used to calculate weights that correspond to the importance of the shot in reconstructing the other shots.

Notice that none of these beforehand cited methods considers fusing data from both mobile and static cameras.

Fig. 9. Example of pedestrian shadows and background object shadows in our database.

Fig. 10. Example of non-rigid deformations in our database.

5. Proposed I2V multi-view pedestrian detection database

This work proposes a new annotated public dataset named Infrastructure to Vehicle Multi-View Pedestrian Detection dataset captured in a real-world environment. In this dataset, multiple sequences were captured from both a static camera fixed in a road infrastructure providing a bird's eye view, and a mobile camera embedded in a car.

5.1. Hardware

The static camera is a GoPro Hero 3, and the mobile camera attached inside the car is a smartphone's camera. Fig. 1 shows the cameras setup. Both mobile and static sequences were captured at a resolution of 1280×720 and at 15 frames per second. The fixed camera has a field of view of 135°, and the mobile camera has a field of view of 107°. To obtain synchronized footage, we first synchronized the clocks of both cameras, and then synchronized the frames based on the timestamps collected from each camera. The synchronization between the fixed and mobile cameras was obtained with ∼30 ms accuracy, the precision of which can be observed in Fig. 2.

5.2. Camera layout and filming locations

Our database's sequences were captured in 4 different locations around the National School of Engineering of Sousse (Tunisia) as outlined in Fig. 3. The fixed camera is mounted on a pole at a height of 5 m, while the mobile camera is mounted inside a car. Both cameras' fields of view are overlapping, although the overlap and its location change in each frame as the car moves.

5.3. Sequence overview

The database contains sixty-one sequences synchronically filmed by the mobile and the static cameras (and four static negative sequences) for a total of 4740 synchronized pairs of frames. The sequences depict a variety of simple and complex scenarios. Table 4 shows samples of these scenarios. Up to ten pedestrians participated in the scenarios. The pedestrians have different heights and wear different clothes and some of them wear accessories such as bags.

5.4. Annotation process

The first 20 sequences were fully annotated at 15 fps, and the other 41 sequences were annotated at a rate of 7.5 fps. Globally, 9480 frames were annotated with a total of 22 127 bounding boxes (11 164 bounding boxes in the static cameras and 10 963 bounding boxes in the mobile cameras). All annotations were performed manually using the MATLAB image labeler app. A sample of the annotations is shown in Fig. 4.

5.5. Challenges

The proposed dataset is meant to be realistic and was designed for real-world applications. Hence, the sequences were filmed in an uncontrolled environment where other pedestrians and cars can enter and exit the filming locations freely. The uncontrolled environment means also that the scene is cluttered with objects such as trees, light poles, signs and other fixtures.

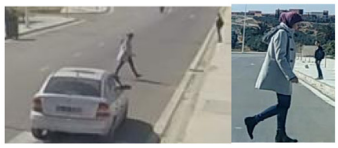
– *Illumination problems* The sequences were filmed in different locations and lighting conditions under natural lighting and the acquisition time varied from midday to afternoon. While some sequences were filmed in sunny conditions, others were filmed in overcast weather and the angle of the sunlight varied depending on the location. These variations created many challenges as shown in Fig. 5. For example, trees and buildings create shadow areas which confuses pedestrian detecting. Moreover, the sunlight causes glare and reflections on the cars' windshield, which affects the detection process in the mobile camera.

– *Appearance variations:* The pedestrians in our database have varied characteristics: There are large variations in body shapes, clothing variations and some pedestrians wear accessories such as backpacks (Fig. 6). The pedestrians are also at different distances and angles from the cameras, which contributes even more to the variations.

– *Occlusions:* Due to the complex backgrounds and various objects present in our filming locations, partial occlusions and full occlusions are frequent in our database: Pedestrians can walk behind trees, bushes, poles or other obstructions, thereby hiding them partially or totally as shown in Fig. 7. Another factor in the occlusion problem is that pedestrians walking in a group often occlude each other. Pedestrians may also be occluded by cars, especially when crossing the road.

– *Complex backgrounds:* Since we chose to film our database in an uncontrolled real-world location, the backgrounds contain elements that increase scenes complexity and may impact the detection process. This is especially true for vertical elements that cause false positives. The backgrounds contain many vertical elements such as trees, poles, signs and some parts of the background buildings (Fig. 8).

Table 4
Different scenarios and their corresponding sequences.

Sequences	Scenarios	Example images from both cameras
1,2,3,4,5,11, 15,16,23,36, 40, 41,56,57	One person crossing the street.	
6,7,10,12, 14,18, 20,21,29, 32, 38, 39,43,58	A group of people crossing the street in same direction, the group might be separated.	
8,9,13,22, 25,33,42,59	2 or more people are crossing the street in different directions simultaneously	
17,19,24,26,27, 28,30,31,34,35, 37, 44,45	2 or more people are crossing the street in different directions at different times	
48,49,50,51, 52,53,54	People are crossing the street, one or more people are occluded from one view by a car	
46,47,55,60, 61	Negative sequence, no pedestrians, cars are driving on the street	
62,63,64,65	Negative sequences, no pedestrians, no car	

– *Shadows:* The illumination conditions described earlier not only cause illumination problems, but also induce shadows of the pedestrians and background objects (Fig. 9). These shadows have different sizes and varying angles and may have similar shapes to pedestrians. These similarities can induce false positives in the detection process.
– *Non-rigid deformations:* Many pedestrian actions naturally impact the body's shape, which increases the detection difficulty. To consider these challenges, we recorded sequences where the pedestrians make sudden movements or direction changes. For example, some sequences include pedestrians that pick up an object on the ground or bend down to tie their shoes as presented in Fig. 10.

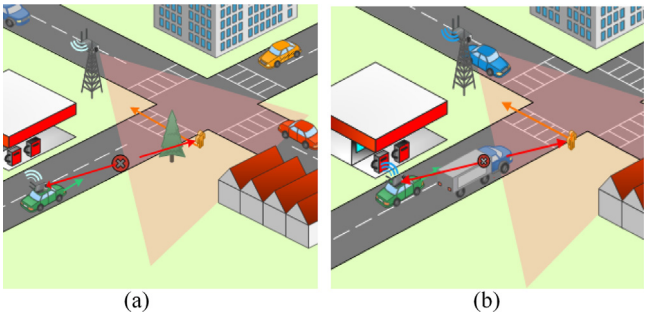
6. Proposed multi-view pedestrian detection framework

The motivation behind the dataset described in Section 5 is to facilitate building robust frameworks of multi-view pedestrian detection that are effective in real-world conditions. In this section, we propose a collaborative intelligence-based perception framework where both roadside infrastructure and vehicle are equipped with cameras (which are assumed to be uncalibrated), processing units, and wireless communication devices. Both fields of view are overlapping. In our framework, the role of the infrastructure camera is to refine the pedestrian detection done by the cameras of nearby vehicle vision systems. Fig. 11 shows motivational scenarios where the collaborative aspects of our framework is beneficial.

The proposed framework is composed of four steps as shown in Fig. 12. The first step is to prepare the monocular detector based on transfer learning techniques using training data from both views. In the next step, we infer our trained detector to each view separately to obtain monocular detection results. To compensate for the camera non-calibration, in the third step, we identify correspondence points between the two views to calculate geometric transformation parameters. Finally, we use these projections to refine the vehicle detection results. We remove the detections with no correspondence in the infrastructure view, and we project the infrastructure view on the vehicle's to update it with the new detections.

Table 5
Major mathematical notations used in this paper.

Mathematical Notation	Definition
I^{Veh}	Input image acquired from the vehicle mounted camera
I^{Inf}	Input image acquired from the infrastructure mounted camera
G^{Veh}	Ground truth annotations of I^{Veh}
G^{Inf}	Ground truth annotations of I^{Inf}
g_k	Ground truth bounding box k in I^{Veh}
g_l	Ground truth bounding box l in I^{Inf}
f	Faster R-CNN detector
BB_k^{Veh}	Detected bounding box k in I^{Veh}
BB_l^{Inf}	Detected bounding box l in I^{Inf}
F_k^{Veh}	Feature vectors obtained from cropped images delimited by BB_k^{Veh}
F_l^{Inf}	Feature vectors obtained from cropped images delimited by BB_l^{Inf}
SM	Similarity matrix between F_k^{Veh} and F_l^{Inf}
BB_n^{Veh}	Best matching bounding box from I^{Veh}
BB_m^{Inf}	Best matching bounding box from I^{Inf}
$CP_{a,b}^{Veh}$	Set of correspondence points from I^{Veh}
$CP_{a,b}^{Inf}$	Set of correspondence points from I^{Inf}
P	Vehicle to infrastructure geometric transformation
P^{-1}	Infrastructure to vehicle geometric transformation



(a) (b)

Fig. 11. Some examples of scenarios encountered in I2V multi-view pedestrian detection applications: (a) A passenger crossing the street is seen by the infrastructure camera but not the car's camera. (b) The car is trying to pass the truck, but it cannot see the passenger, but infrastructure camera can.

In the following sections, we detail the different steps of our proposed framework. For the sake of clarity, we introduce the major mathematical notations in Table 5.

We split our database in two sets: the training DB_{train} (Eq. (1)) and validation DB_{test} (Eq. (2)) databases:

$$DB_{train} = \left\{\left(I^{Veh}, G^{Veh}\right)_i; \left(I^{Inf}, G^{Inf}\right)_i\right\}_{i=1}^{|DB_{train}|} \quad (1)$$

$$DB_{test} = \left\{\left(I^{Veh}, G^{Veh}\right)_i; \left(I^{Inf}, G^{Inf}\right)_i\right\}_{i=1}^{|DB_{test}|} \quad (2)$$

where $G^{Veh} = \{g_k\}_{k=1}^{|G^{Veh}|}$ and $G^{Inf} = \{g_l\}_{l=1}^{|G^{Inf}|}$

with $g_k = \left[x_k^g, y_k^g, w_k^g, h_k^g\right] \in \mathbb{R}^4$ and $g_l = \left[x_l^g, y_l^g, w_l^g, h_l^g\right] \in \mathbb{R}^4$ denoting the top left corner, width and height of the ground truth bounding box respectively.

6.1. Transfer learning

In our framework, we propose a deep learning-based approach using various pre-trained models that we adapt to our context by transfer learning techniques (Fig. 12.A) More precisely, we opt for Faster R-CNN architectures that are trained using data from DB_{train}. This choice is motivated by Faster R-CNN ability to achieve a good compromise between performance and speed in various applications [54,55].

Each of input images I^{Veh} and I^{Inf} goes through a CNN which has two tasks: extract the convolutional feature maps of the image, and propose the potential object detection regions. The regions are proposed around anchor points that are selected via a sliding window. The region proposal network (RPN) chooses the region size in different scales and aspect ratios. To deploy our Faster R-CNN detector we fine tune a pre-trained detector and adapt it to our task. This method is much faster than training a network from scratch and requires less training data to achieve good results. Faster R-CNN fine tuning is performed by back-propagation and the stochastic gradient descent method in 4 steps [54]:

i. Fine tune the pre-trained model's RPN using the training data
ii. Fine tune a separate detection network using the RPN from step i.
iii. Re-train the RPN while sharing the weights from the detector trained in step ii. (The weights of the detection networks are kept fixed, only the RPN weights change)
iv. Re-train the network using the RPN obtained from the previous step.

Once the training is complete, we obtain a detector f that detects pedestrians within the image I^{Veh} or I^{Inf}.

6.2. Monocular detection

At a certain time t, both the static camera and the mobile camera acquire a pair of images (I^{Inf}, I^{Veh}). In this step, we run monocular pedestrian detection on each image of the pair using our trained detector f (Fig. 12.B). The output of this step is two sets of detections $BB_l^{Inf} = f(I^{Inf})$ and $BB_k^{Veh} = f(I^{Veh})$ for each image:

- For the infrastructure image, we obtain the detections $BB_l^{Inf} = \{[x_l, y_l, w_l, h_l]\}_{l=1}^{|BB_l^{Inf}|}$, where:
  - $[x_l, y_l, w_l, h_l]$ are the parameters of the bounding boxes, with (x_l, y_l) being the top left corner of bounding box l, w_l and h_l its width and height, respectively.

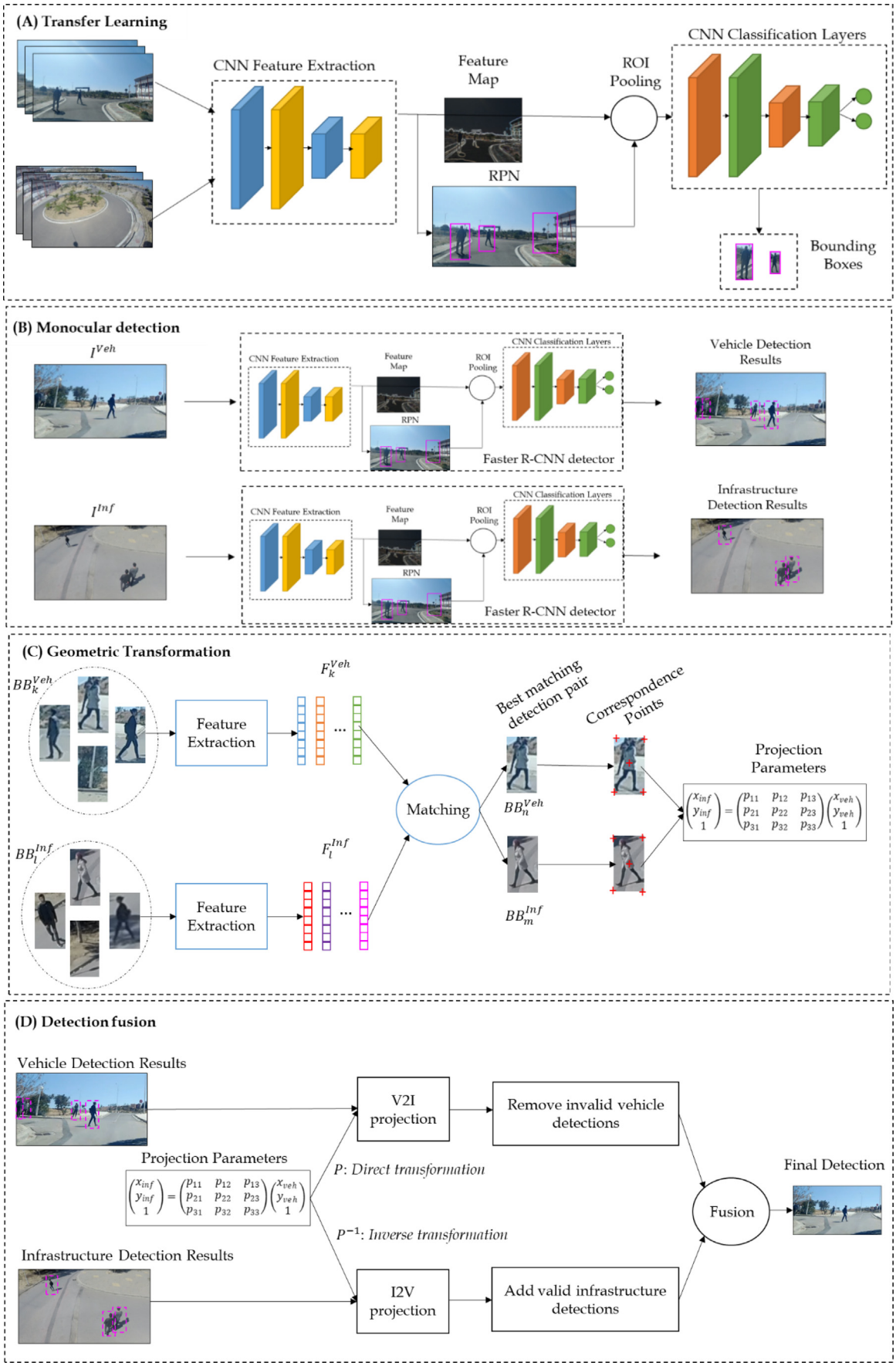


Fig. 12. Illustration of the key steps of our proposed framework.

- For the vehicle image, we obtain the detections $BB_k^{Veh} = \{[x_k, y_k, w_k, h_k]\}_{k=1}^{|BB_k^{Veh}|}$, where:
  - $[x_k, y_k, w_k, h_k]$ are the parameters of the bounding boxes, with (x_k, y_k) being the top left corner of bounding box k, w_k and h_k its width and height, respectively.

6.3. Geometric transformation

After obtaining the detection results BB_l^{Inf} and BB_k^{Veh}, we implement a collaborative mechanism to detect potentially occluded pedestrians and overcome the limitations of monocular detection. The decision making is then obtained from a collaboration between both infrastructure and vehicle's intelligent systems. However, the challenge in our case is the unavailability of camera calibration data and consequently the impossibility to apply common ground plane projection and back-projection methods. To solve this problem, we propose a transformation that projects points from either view to the other one. This method has three stages: First, we find a pair of bounding boxes that contain the same person, then we use this information to obtain interest points that are finally used to calculate projection parameters between the two views (Fig. 12.C).

• Finding the matching bounding box pair

To find similar points in both images, we look for similar regions in the images. An intuitive solution is a pedestrian that appears in both images I^{Inf} and I^{Veh}. In this case, finding similar regions consists of finding the pair of bounding boxes in BB_l^{Inf} and BB_k^{Veh} that correspond to the same pedestrian. First, pedestrian bounding box features are extracted, and then similarity scores between vehicle side detections and infrastructure side detections are calculated. The pair with the best matching score is retained.

For the feature extraction process, we calculate the feature vectors F_l^{Inf} and F_k^{Veh} of the cropped images delimited by the bounding boxes $I^{Inf}(BB_l^{Inf})$ and $I^{Veh}(BB_k^{Veh})$. To this aim, we evaluate four commonly used feature extraction methods: Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), the Gaussian of Gaussian (GOG) method and the Learned Features (LF) [1,56–58].

- In the HOG method [56], the image is first divided into square cells, and overlapping blocks are created using the cells. In each cell, a histogram of gradient intensities is calculated in a bin containing a number of directions. Then, cell histograms are normalized in each block, and the cell histograms are concatenated to form the final feature vector.
- In the LBP method [57], a number of neighboring pixels is selected around a radius of the center for each pixel, and the sign of the difference in intensity between the center pixel and each of its neighbors is calculated. If the sign is positive, the neighboring pixel is given a value of 1, and 0 otherwise. These values are then concatenated to form a binary pattern. The final feature vector is a histogram of the occurrences of these patterns.
- In the GOG method [58], each image is divided in overlapping regions. Each region is subsequently divided into patches. First, pixel features such as the pixel's vertical position, the magnitude of the intensity gradient in four directions and the RGB color channel values, etc. are extracted. Patch features are extracted by summarizing the pixel features with a Gaussian distribution, and a similar distribution is used to obtain region features from patch features. Pixel, Patch and Region features are then concatenated to form the feature vector.
- Learned features are automatically obtained by training a Deep Convolutional Neural Network architecture on a labeled dataset [1]. Different types of architectures can be used such as VGG-16, VGG-19, Resnet 18, Mobilenetv2, etc. Generally, the learned features are extracted after the last convolution layer of the models. The learned features are then forwarded to a classifier (ANN, k-NN, SVM...).

After feature extraction, we obtain a feature vector for each bounding box:

$$F_l^{Inf} = \begin{cases} HOG\left(I^{Inf}(BB_l^{Inf})\right) \\ or \\ LBP\left(I^{Inf}(BB_l^{Inf})\right) \\ or \\ GOG\left(I^{Inf}\left(BB_l^{Inf}\right)\right) \\ or \\ LF\left(I^{Inf}(BB_l^{Inf})\right) \end{cases}, \; l = 1..|BB_l^{Inf}| \tag{3}$$

$$F_k^{Veh} = \begin{cases} HOG\left(I^{Veh}(BB_k^{Veh})\right) \\ or \\ LBP\left(I^{Veh}(BB_k^{Veh})\right) \\ or \\ GOG\left(I^{Veh}(BB_k^{Veh})\right) \\ or \\ LF\left(I^{Veh}(BB_k^{Veh})\right) \end{cases}, \; k = 1..|BB_k^{Veh}| \tag{4}$$

These feature vectors are used to calculate a similarity matrix SM between F_l^{Inf} and F_k^{Veh}. This matrix is defined as:

$$SM = (d_{kl}), \; k = 1..\left|BB_k^{Veh}\right|, \; l = 1..\left|BB_l^{Inf}\right| \tag{5}$$

where:

$$d_{kl} = dist\left(F_k^{Veh}, F_l^{Inf}\right), \; k = 1..|BB_k^{Veh}|, \; l = 1..|BB_l^{Inf}| \tag{6}$$

With $dist$ representing the distance between the feature vectors.

The matching pair is the pair $\left(BB_n^{Veh}, BB_m^{Inf}\right)$, where (m, n) are the indices that satisfy the following equation:

$$d_{mn} = \min(SM) \tag{7}$$

• Finding the correspondence points

Using the pair of matched bounding boxes $BB_n^{Veh} = [x_n, y_n, w_n, h_n]$ and $BB_m^{Inf} = [x_m, y_m, w_m, h_m]$, we are able to find correspondence points between the infrastructure view and the vehicle view. To obtain the best projection parameters, we empirically select the points on a 9×9 grid placed on top of each of the matched bounding boxes. We obtain two sets of correspondence points CP^{Inf} and CP^{Veh} given by the following equations:

$$CP_{a,b}^{Inf} = \left(x_m + w_m \times \frac{a}{8}, y_m + h_m \times \frac{b}{8}\right), a = 0 \ldots 8, b = 0 \ldots 8 \tag{8}$$

$$CP_{a,b}^{Veh} = \left(x_n + w_n \times \frac{a}{8}, y_n + h_n \times \frac{b}{8}\right), a = 0 \ldots 8, b = 0 \ldots 8 \tag{9}$$

As expressed in Eq. (10), the two sets of points are matched such that points on the same position on the grid match together:

$$CP_{a,b}^{Inf} \Leftrightarrow CP_{a,b}^{Veh}, a = 0 \ldots 8, b = 0 \ldots 8 \quad (10)$$

• Calculating the projection parameters

In the last stage, we have obtained 81 pairs of corresponding points. We can now use these points to estimate projection parameters between our pair of images I^{Inf} and I^{Veh}. To obtain these parameters, we need to solve the following equations:

$$\begin{pmatrix} x_{a,b}^{inf} \\ y_{a,b}^{inf} \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} x_{a,b}^{Veh} \\ y_{a,b}^{Veh} \\ 1 \end{pmatrix}, \quad a = 0 \ldots 8, b = 0 \ldots 8 \quad (11)$$

where $(x_{a,b}^{inf}, y_{a,b}^{inf})$ and $(x_{a,b}^{Veh}, y_{a,b}^{Veh})$ are a pair of corresponding points from the $CP_{a,b}^{Inf}$ and $CP_{a,b}^{Veh}$ point sets we obtained earlier. Once this step is done, we obtain two transformation matrices: P for the vehicle to infrastructure geometric transformation, and P^{-1} for the infrastructure to vehicle geometric transformation.

6.4. Detection fusion

In the last step, we have obtained the transformations between the infrastructure view and the vehicle view, which will allow both the infrastructure side system and the vehicle side system to collaborate in order to refine the vehicle camera's detection results. This is done in two steps: First, we remove detections from the vehicle view that do not have a projection in the infrastructure view. Then, we augment the remaining detections from last step with projected detections from the infrastructure view (Figure II-13 D), which leads to an improved perception capability of the vehicle's system.

• Removing vehicle view detections with invalid projections

To remove detections that do not have a corresponding projection in the infrastructure image, we follow Algorithm 1:

After Algorithm 1 is executed, we obtain a reduced set of detections.

• Adding the projected infrastructure detections

To add the detections of the infrastructure camera to the detections we obtained in the previous step, we project the bounding boxes from the infrastructure view to the vehicle view and verify that they are valid additions following Algorithm 2.

Once this step is done, we obtain the final detection results, which are the added projected bounding boxes combined with the reduced set of detections from the previous step. This final detection is the result of the collaborative intelligence between the infrastructure system and the vehicle perception system.

7. Experiments and results

All the experiments were performed on a PC with an Intel (R) core (TM) i7-9700H CPU @ 2.60 GHz, 16 GB of RAM, and an NVIDIA GTX 1050Ti graphics card with 4 GB of VRAM.

7.1. Performance metrics

To evaluate the performance of our method, we used traditional metrics (Precision, Recall, Average Precision) as well as the CLEAR metrics (MODA and MODP) [59]. To calculate these metrics for each method, we measure the number of True Positives (TP), False Negatives (FN) and False Positives (FP).

Algorithm 1:

i. Calculate the coordinates of the centers $\left(xc_l^{inf}, yc_l^{inf}\right), l = 1 \ldots |BB_l^{Inf}|$ of the infrastructure detection bounding boxes:

$$xc_l^{inf} = x_l + \frac{w_l}{2}, l = 1 \ldots |BB_l^{Inf}| \quad (12)$$

$$yc_l^{Inf} = y_l + \frac{h_l}{2}, l = 1 \ldots |BB_l^{Inf}| \quad (13)$$

ii. ***For*** $k = 1$ *to* $|BB_k^{Veh}|$

Calculate the coordinates of the center of BB_k^{Veh}

$$xc_k^{Veh} = x_k + \frac{w_k}{2} \quad (14)$$

$$yc_k^{Veh} = y_k + \frac{h_k}{2} \quad (15)$$

iii. Project this center to the infrastructure view:

$$[xc_k^{Pro}, yc_k^{Pro}, 1]^T = P \times \left[xc_k^{Veh}, yc_k^{Veh}, 1\right]^T \quad (16)$$

With (xc_k^{Pro}, yc_k^{Pro}) the coordinates of the projected point.

iv. Calculate the distances δ_l between the projected point (xc_k^{Pro}, yc_k^{Pro}) and the centers calculated in step i:

$$\delta_l = \sqrt{\left(xc_k^{Pro} - xc_l^{inf}\right)^2 + \left(yc_k^{Pro} - yc_l^{inf}\right)^2} \quad (17)$$

v. Check if the projected center (xc_k^{Pro}, yc_k^{Pro}) is within a certain distance Δ of the infrastructure bounding box centers:

If $\delta_l > \Delta$, the detection BB_k^{Veh} is removed, otherwise, it is kept.

End For

Algorithm 2:

i. ***For*** $l = 1$ *to* $|BB_l^{Inf}|$

Project both the top left corner and bottom right corner of BB_l^{Inf} to the vehicle view:

$$\left[xt_l^{pro}, yt_l^{pro}, 1\right]^T = P^{-1} \times [x_l, y_l, 1]^T \tag{18}$$

$$\left[xb_l^{pro}, yb_l^{pro}, 1\right]^T = P^{-1} \times [x_l + w_l, y_l + h_l, 1]^T \tag{19}$$

ii. Check if both projected corners are within the vehicle image bounds, if one of the projected points is out of bounds, the projection is rejected.

iii. Check if the projected bounding box does not overlap with the pre-existing bounding boxes, if the projected bounding box overlaps with a ratio higher than a certain overlap threshold, it is rejected.

iv. If the projected bounding box satisfies both conditions described in steps ii and iii, it is added to the final detection results.

End For

- *Precision*: This metric measures the ratio of correct detections within the total number of detections.

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

- *Recall*: This metric measures the ratio of correct detections compared to the total number of ground truths.

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

- *Average precision*: This metric gives a quick overview of the detector's performance at every recall level. It is obtained by calculating the area under the precision–recall curve.
- *Multiple Object Detection Accuracy (MODA)*: This is a normalized metric that better evaluates the detector's accuracy by penalizing both false positives and missed detections. MODA is calculated for a single frame t as:

$$MODA\,(t) = 1 - \frac{c_m \times MD + c_f \times FP}{NG} \tag{22}$$

where:

  - MD is the number of missed detections.
  - c_m is a weight that is used to adjust the penalty for missed detections.
  - FP is the number of false positives.
  - c_f is a weight that is used to adjust the penalty for false positives.
  - NG is the number of ground truths.

To evaluate a detector's accuracy on multiple images, a normalized score is calculated:

$$MODA = 1 - \frac{\sum_{t=1}^{N} \left(c_m \times MD\,(t) + c_f \times FP\,(t)\right)}{\sum_{t=1}^{N} (NG\,(t))} \tag{23}$$

where:

  - N is the total number of images.

- *Multiple Object Detection Precision (MODP)*: This metric evaluates the precision of correctly detected objects position by calculating the overlap between each correct detection and its associated ground truth. To calculate MODP, first the overlap between mapped detections and ground truths in a frame is calculated by determining the intersection over union of each pair bounding boxes:

$$Mapped\ overlap\ ratio\ (MOR) = \sum_{i=1}^{N_m} \frac{G_i \cap D_i}{G_i \cup D_i} \tag{24}$$

where:

  - N_m is the number of mapped detections.
  - G_i is the bounding box of ground truth i.
  - D_i is the bounding box of detection i.

To obtain MODP for a single frame, MOR is normalized by dividing it by the number of ground truths:

$$MODP\,(t) = \frac{MOR}{N_m} \tag{25}$$

To evaluate a detector's bounding box placement precision on multiple images, a normalized score is calculated:

$$MODP = \frac{\sum_{i=1}^{N_{frames}} MODP\,(t)}{N_{frames}} \tag{26}$$

where:

  - N_{frames} is the total number of images.

7.2. Pre-trained CNN architectures and adaptation

To train the detectors, we use a subset of our database as training data: We use 20 fully annotated sequences as our training data, which corresponds to 3862 images out of the 9480 in the I2V-MVPD database. As mentioned in Section 6.1, we use a pre-trained CNN architecture and adapt it to our application with transfer learning. In the following experiments, we use different architectures given in Table 6.

All of these models have been pre-trained on the ImageNet[4] database, which contains more than a million images divided in 1000 classes. However, in our case we only have two classes: the presence or absence of a pedestrian. Therefore, architectural modifications of these architectures were performed. These modifications consist of changing the last fully connected layer to only have two outputs and changing the following *soft-max* layer

[4] http://www.image-net.org.

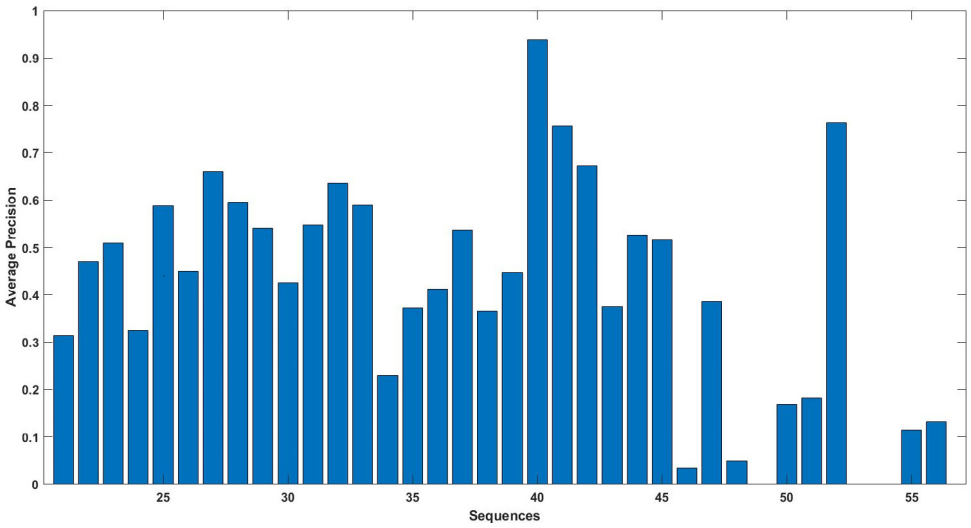


Fig. 13. Mobilenetv2 per sequence average precision for the infrastructure view.

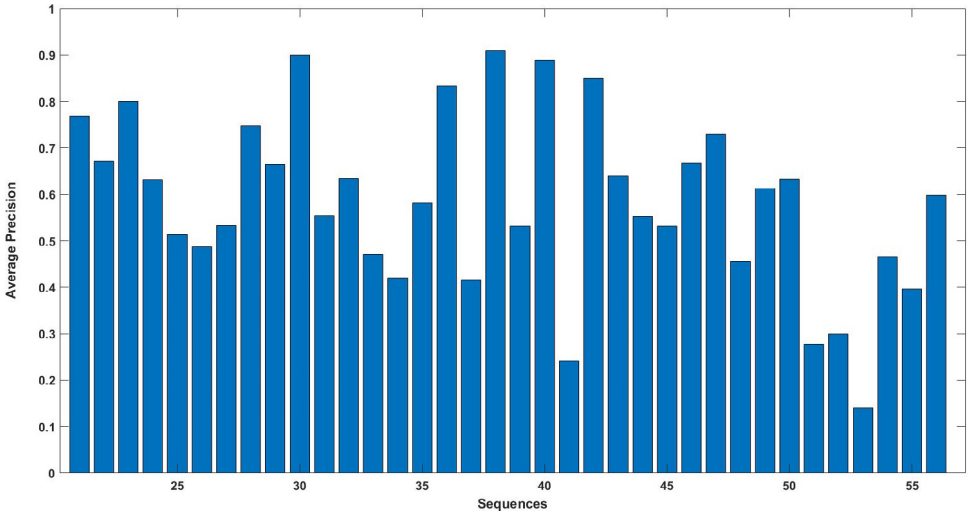


Fig. 14. Mobilenetv2 per sequence average precision for the vehicle view.

Fig. 15. An example of the detector not detecting far pedestrians. Ground truths are in blue and detections are in pink. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to cope with the new fully connected layer. After this adaptation of the pre-trained CNN models, we re-train (fine-tune) the Faster R-CNN detector using our training database. The following hyperparameters were used (Table 7):

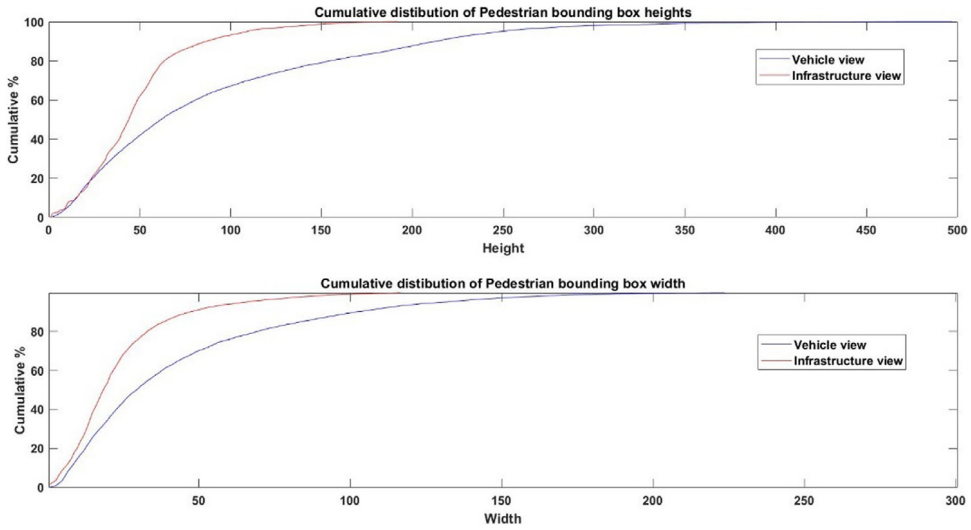


Fig. 16. Cumulative distribution of bounding box sizes in both views. Vehicle sizes are in blue and infrastructure sizes are in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Different CNN architectures used in our experiments.

Architecture	Layers	Number of parameters (millions)
VGG-16	16	138
VGG-19	19	144
GoogleNet	22	7
Resnet 18	18	11.7
Mobilenetv2	54	3.5

Table 7
Faster R-CNN training hyperparameters.

Parameter	Value
Mini Batch Size	1-4[a]
Learn Rate	10^{-4}
Momentum	0.9
Maximum number of epochs	20
Number of Strongest Regions	2000
Number of Regions To Sample	128

[a]The mini batch size depends on the size of the Faster R-CNN network and on hardware limitations

7.3. Monocular detection evaluation

In this section, we evaluate the performance of the trained detectors in the monocular detection on both the infrastructure camera and the vehicle camera. The evaluated detectors are the ACF (Aggregate Channel Features) detector, and Faster R-CNN detectors based on five different architectures: VGG-16, VGG-19, GoogleNet, Resnet 18, Mobilenetv2. The detection results for the infrastructure view and the vehicle view are shown in Table 8 and Table 9, respectively.

The results show that the Faster R-CNN model based on the Mobilenetv2 network has the best detection results while also being the fastest CNN. However, his timing (around 2 fps) is still too slow for real-time detection therefore more optimization is needed to achieve better speeds. The inefficient MATLAB implementation of the CNN framework also contributes to this problem. We also notice that while the ACF based detector has poor detection results, especially in the precision aspect, its detection time is much faster than that of the CNN based detectors and is almost able to reach real-time speeds.

Another noticeable aspect is that vehicle view detection results are better than those of the infrastructure view. This is mainly due to the vehicle being closer to pedestrians than the infrastructure camera, which makes them easier to detect. In fact, the average pedestrian bounding box size in the vehicle view is 114×50 pixels, while its infrastructure counterpart is 62×29 pixels.

Fig. 17. An example of the detector failing to detect occluded pedestrians. Ground truths are in blue and detections are in pink. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 18. An illustration of false positives. Ground truths are in blue and detections are in pink. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 8
Detection performance for the infrastructure view.

	ACF	VGG-16	VGG-19	GoogleNet	Resnet 18	Mobilenetv2
Average Precision	23.68	44.10	26.37	30.78	25.68	**44.01**
True Positives	2565	3507	2694	2908	28.59	**3561**
False Positives	8211	6539	6343	3389	7636	**2076**
False Negatives	4117	3175	3988	3774	3823	**3121**
Precision	23.80	34.91	29.81	46.18	27.24	**63.17**
Recall	38.39	52.48	40.32	43.52	42.79	**53.29**
MODA	−0.844	−0.454	−0.546	−0.072	−0.715	**0.222**
MODP	0.377	0.575	0.406	0.493	0.456	**0.547**
Time (s)	0.037	0.745	0.785	1.867	1.142	**0.568**

Table 9
Detection performance for the vehicle view.

	ACF	VGG-16	VGG-19	GoogleNet	Resnet 18	Mobilenetv2
Average Precision	27.37	50.51	50.48	50.65	47.16	**55.07**
True Positives	2712	3550	3383	3522	3427	**3674**
False Positives	14581	7407	2717	4686	6933	**2730**
False Negatives	3213	2375	2542	2403	2498	**2251**
Precision	15.68	32.40	55.46	42.91	33.08	**57.37**
Recall	45.77	59.92	57.10	59.44	57.84	**62.01**
MODA	−2.003	−0.651	0.112	−0.196	−0.591	**0.159**
MODP	0.394	0.497	0.595	0.606	0.511	**0.606**
Time (s)	0.035	0.760	0.778	1.764	1.190	**0.487**

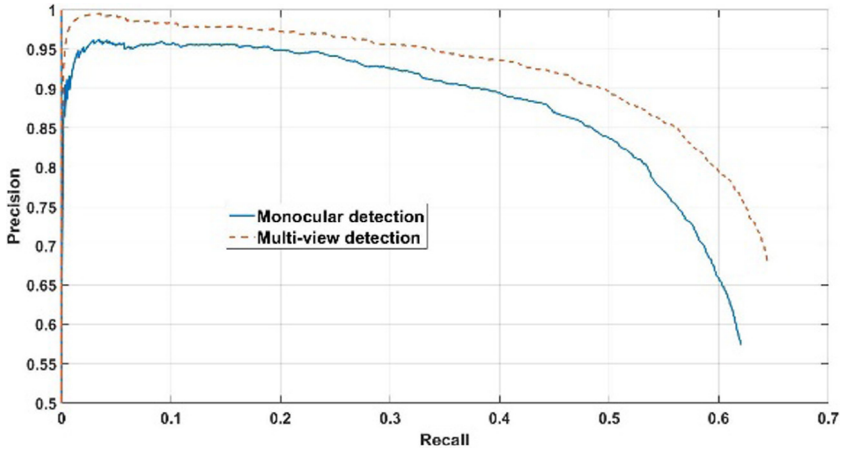


Fig. 19. Precision–recall curves of monocular detection and our proposed multi-view framework.

An analysis of the per sequence performance gives further insight into the strengths and weaknesses of our detectors. Figs. 13 and 14 show the per sequence average precision of the best performing detector (Mobilenetv2).

We notice that the detectors perform well in normal conditions, where the pedestrians tend to be closer to the camera and have no obstructions impeding the detection. However, performance drops when pedestrians are far away from the camera. An illustration of this is shown in Fig. 15, which is taken from sequence 34 of the infrastructure view, where the two nearest pedestrians are detected consistently, and the two farthest pedestrians were not detected. This corresponds to a detection distance of approximately 60 m.

This weakness in detecting far pedestrians is a challenge in our database, as many pedestrian bounding boxes are small. Fig. 16 shows a cumulative distribution of pedestrian bounding box height and width in both views: 25.5% of the vehicle view pedestrians and 28% of the infrastructure view pedestrians had a height of 30 pixels or less, which is difficult to detect even for human annotators according to [60].

Another weakness of the monocular detectors is their occlusion handling. In situations where people move in groups, the occluded pedestrians are rarely detected (Fig. 17), and pedestrians partially occluded by objects in the environment face similar issues.

Another problem encountered in the detection process is the high number of false positives. Trees and signs are common false positive detections, as well as certain elements on the vehicles and some shadows. This problem is more pronounced in the vehicle view, as shown by its lower precision score. Fig. 18 shows illustrations of false positives.

7.4. Multi-view detection evaluation

In the following, we evaluate the performance of our multi-view pedestrian detection framework and quantify the improvements made to the results.

7.4.1. Parameter settings

For the geometric transformation step, we conduct a comparative study in order to define the parameter settings of our proposed framework. These parameters are: the feature extraction method applied to the cropped images, the distance used in the matching stage and the projection method.

Fig. 20. An example of false positive removal. Monocular detections are on the left and multi-view detections are on the right.

Table 10 shows the effect of the different feature extraction methods (LBP, HOG, GOG and LF). For LF, we evaluated several pre-trained models, the best results are achieved with the mobilenetv2, pre-trained on the ImageNet database. In our case, as reported in [61] for image classification on other datasets, we remove the last fully-connected layer, and use 1280-D activations of the penultimate layer as image features. We observe that LBP is the best performing feature extraction method and will be adopted for the remaining stages. The other methods (HOG and GOG) have comparable performance results. While LF-based classification gives the lowest performance, it takes more than 4 times the processing time of the LBP-based system. The better results given by hand-crafted methods is explained by the low resolution of post-cropping images.

The next parameter is the distance used in the matching stage. The Euclidean distance has shown the best results as seen in Table 11 and will be adopted for the next stage, while the other distances have similar results. Time-wise, the Euclidean distance is slightly slower than the other distances, with an additional 14 ms. This step is important since this will determine the detections from both views to match, and a wrong match causes a faulty projection.

Table 10
Detection performance for the different feature extraction methods.

	LBP	HOG	GOG	LF
Average Precision	**59.79**	54.24	54.19	51.04
True Positives	**3820**	3627	3637	35.17
False Positives	**1814**	2106	2143	2142
False Negatives	**2105**	2298	2288	2408
Precision	**67.80**	63.27	62.92	60.78
Recall	**64.47**	61.21	61.38	58.05
MODA	**0.339**	0.256	0.252	0.232
MODP	**0.606**	0.595	0.598	0.588
Time (s)	**0.642**	0.629	4.054	2.741

Table 11
Detection performance for the different matching distances.

	Euclidean	Cityblock	Chebychev	Correlation	Spearman
Average Precision	**59.79**	55.25	54.73	55.11	53.99
True Positives	**3820**	3691	3675	3704	3617
False Positives	**1814**	2148	2171	2169	2114
False Negatives	**2105**	2234	2250	2221	2308
Precision	**67.80**	63.21	62.86	63.07	61.05
Recall	**64.47**	62.30	62.02	62.51	63.11
MODA	**0.339**	0.260	0.253	0.259	0.254
MODP	**0.606**	0.598	0.599	0.600	0.593
Time (s)	**0.642**	0.628	0.628	0.629	0.656

Table 12
Detection performance for the different types of projections.

	Projective	Affine	Similarity	Polynomial order 4	Polynomial order 3	Polynomial order 2
Average Precision	**59.79**	54.45	51.46	37.50	46.26	54.58
True Positives	**3820**	3702	3706	2533	3271	3680
False Positives	**1814**	2311	2912	1634	1969	2189
False Negatives	**2105**	2223	2219	3392	2654	2245
Precision	**67.80**	61.57	56.00	60.79	62.42	62.70
Recall	**64.47**	62.48	62.55	42.75	55.20	62.10
MODA	**0.339**	0.235	0.134	0.152	0.220	0.252
MODP	**0.606**	0.600	0.599	0.566	0.580	0.601
Time (s)	**0.642**	0.657	0.658	0.651	0.649	0.652

Fig. 21. Occluded pedestrians are added to the detections (in light blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The final parameter is the type of projection. As shown in Table 12, the Projective method has the best results. These results can be explained by the flexibility of the projective method. Another noticeable aspect of the results is the poor performance of polynomial transforms. These results get significantly worse as their order gets higher. Projective method is also slightly faster than the other methods.

Table 13
Performance difference between monocular detection and our proposed multi-view framework.

	Monocular	Multi-view	Improvement
Average Precision	55.07	59.79	+4.72%
True Positives	3674	3820	+146
False Positives	2730	1814	-916
False Negatives	2251	2105	-146
Precision	57.37	67.80	+10.43%
Recall	62.01	64.47	+2.46%
MODA	0.159	0.339	+0.180
MODP	0.606	0.606	+0
Time (s)	0.487	0.642	+0.155

To summarize, we adopt the LBP feature extraction method, the Euclidean distance for the matching stage and the Projective method in our finalized framework.

7.4.2. Quantitative and qualitative evaluation

The performance improvements between monocular detection and our I2V multi-view pedestrian detection framework are shown in Table 13 and in Fig. 19.

Our method has shown significant improvement over the monocular results: There is a slight improvement in the number of correct detections, and a significant improvement in reducing the number of false positives. A third of the false positives was removed by our method, which led to a significant precision improvement. Moreover, MODA score is also improved; it has been doubled with our method. However, our method slows down detection. While this time overhead is not significant, optimizations are to be deployed to achieve faster detection systems.

Our method eliminates a large number of false positives, especially if they happen to be away from the main group of pedestrians. These removed false positives fix the issue encountered in monocular detection shown in Fig. 18. However, our method may fail to remove false positives located near correct detections. Fig. 20 shows a sample of false positive removal.

Another contribution of our method is to correctly detect partially occluded pedestrians by projecting them from the infrastructure view as shown in Fig. 21.

Moreover, the problem of detecting distant pedestrians in the monocular detection has been successfully addressed by our

Fig. 22. Distant pedestrians are detected by our method (in light blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

method. Since a pedestrian can be far from a view and close to another view, the detection can be achieved by projecting the detection from the close view to the far view. Fig. 22 shows an example of such a case.

7.5. Discussion

Nowadays transportation systems are undergoing disruptive transformations led by various technologies such as connected cars and artificial intelligence-driven technologies. These transformations are human-centric and try to afford the maximal user satisfaction with the highest possible safety guarantees. Since pedestrians are the most vulnerable road users, reliably detecting them is a crucial task. In this work, we propose a novel public dataset that considers a collaborative vehicle-infrastructure setting where both sides are equipped with intelligent systems.

To demonstrate the interest of our dataset, a new multi-view pedestrian detection framework based on collaborative intelligence was presented. Our results show a significant improvement in detection performance over monocular detection. Nevertheless, a considerable false positives rate was recorded. In fact, our system is still partially confusing human-like obstacles with pedestrians. While this aspect needs further improvements, it is worth noticing that our method eliminated more than 33% of false positives of a monocular system.

In our framework, we chose to set a generic case where the whole camera view is considered. However, scenes that are too far from the vehicle, or from the road may be neglected in a preprocessing phase without undermining safety. The detection accuracy can thereby be enhanced by simply restricting the region of interest.

We believe that the collaborative aspect offered by the proposed dataset could be exploited in a plethora of research directions. For example, while the low resolution of the bounding boxes is challenging for keypoints matching, specialized CNNs might be used as descriptors after keypoints are detected through a comprehensive technique. In another promising direction, researchers may develop reinforcement learning techniques based on the different parts' feedback. Moreover, federated and distributed-learning approaches in the ITS context may be benchmarked. We also believe that contributions to tackle machine-learning security problems such as adversarial attacks may rely on the collaboration between the infrastructure and the vehicle.

8. Conclusion

In this paper, we present a novel pedestrian detection database that combines synchronized images from both an infrastructure side camera and an in-vehicle embedded camera for intelligent transportation systems. To further illustrate the relevance of the dataset, we also propose a new multi-view pedestrian detection framework that fuses detection even without calibration data. The results showed a clear improvement over monocular detection in most metrics. We believe that this study could be a valuable input for future development of real-time collaborative perception systems in which intelligent infrastructure can communicate and assist intelligent vehicles.

CRediT authorship contribution statement

Anouar Ben Khalifa: Conceptualization, Writing - original draft, Software, Methodology, Visualization, Investigation, Validation. **Ihsen Alouani:** Conceptualization, Methodology, Writing - review & editing, Validation. **Mohamed Ali Mahjoub:** Supervision, Writing - review & editing. **Atika Rivenq:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Yuan Wang, Dongxiang Zhang, Ying Liu, Bo Dai, Loo Hay Lee, Enhancing transportation systems via deep learning: A survey, Transp. Res. C 99 (2019) 144–163, http://dx.doi.org/10.1016/j.trc.2018.12.004.

[2] Wei Xiang, Tao Huang, Wanggen Wan, Machine learning based optimization for vehicle-to-infrastructure communications, Future Gener. Comput. Syst. 94 (2019) 2019, http://dx.doi.org/10.1016/j.future.2018.10.047.

[3] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, Vitoantonio Bevilacqua, Computer vision and deep learning techniques for pedestrian detection and tracking: A survey, Neurocomputing 300 (2018) 17–33, http://dx.doi.org/10.1016/j.neucom.2018.01.092.

[4] Cristina Olaverri-Monreal, Gerd Ch. Krizek, Florian Michaeler, Rene Lorenz, Matthias Pichler, Collaborative approach for a safe driving distance using stereoscopic image processing, Future Gener. Comput. Syst. 95 (2019) 2019, http://dx.doi.org/10.1016/j.future.2018.01.050.

[5] Felipe Jiménez, José Eugenio Naranjo, José Javier Anaya, Fernando García, Aurelio Ponz, José María Armingol, Advanced driver assistance system for road environments to improve safety and efficiency, Transp. Res. Procedia 14 (2016) 2245–2254, http://dx.doi.org/10.1016/j.trpro.2016.05.240.
[6] Qiangqiang Guo, Li Li, Xuegang (Jeff) Ban, Urban traffic signal control with connected and automated vehicles: A survey, Transp. Res. C 101 (2019) 313–334, http://dx.doi.org/10.1016/j.trc.2019.01.026.
[7] Anouar Ben Khalifa, Ihsen Alouani, Mohamed Ali Mahjoub, Najoua Essoukri Ben Amara, Pedestrian detection using a moving camera: A novel framework for foreground detection, Cogn. Syst. Res. (2019) http://dx.doi.org/10.1016/j.cogsys.2019.12.003.
[8] Amira Mimouna, Ihsen Alouani, Anouar Ben Khalifa, Yassin El Hillali, Abdelmalik Taleb-Ahmed, Atika Menhaj, Abdeldjalil Ouahabi, Najoua E. Ben Amara, OLIMP: A heterogeneous multimodal dataset for advanced environment perception, Electronics 9 (4) (2020) 560, http://dx.doi.org/10.3390/electronics9040560.
[9] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, Homayoun Najjaran, Autonomous vehicle perception: The technology of today and tomorrow, Transp. Res. C 89 (2018) 384–406, http://dx.doi.org/10.1016/j.trc.2018.02.012.
[10] M. Goldhammer, E. Strigel, D. Meissner, U. Brunsmann, K. Doll, K. Dietmayer, Cooperative multi sensor network for traffic safety applications at intersections, in: 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, 2012, pp. 1178–1183, http://dx.doi.org/10.1109/ITSC.2012.6338672.
[11] Sabri M. Hanshi, Tat-Chee Wan, Mohammad M. Kadhum, Ali Abdulqader Bin-Salem, Review of geographic forwarding strategies for inter-vehicular communications from mobility and environment perspectives, Veh. Commun. 14 (2018) 64–79, http://dx.doi.org/10.1016/j.vehcom.2018.09.005.
[12] S. Vishnu, U. Ramanadhan, N. Vasudevan, A. Ramachandran, Vehicular collision avoidance using video processing and vehicle-to-infrastructure communication, in: International Conference on Connected Vehicles and Expo, ICCVE, Shenzhen, 2015, pp. 387–388, http://dx.doi.org/10.1109/ICCVE.2015.36.
[13] M.A. García-Garrido, M. Ocaña, D.F. Llorca, E. Arroyo, J. Pozuelo, M. Gavilán, Complete vision-based traffic sign recognition supported by an I2V communication system, Sensors 12 (2012) 1148–1169, http://dx.doi.org/10.3390/s120201148.
[14] I. Jegham, Anouar Ben Khalifa, I. Alouani, M.A. Mahjoub, Safe driving: Driver action recognition using SURF keypoints, in: 30th International Conference on Microelectronics, ICM, Sousse, Tunisia, 2018, pp. 60–63, http://dx.doi.org/10.1109/ICM.2018.8704009.
[15] Imen Jegham, Anouar Ben Khalifa, Ihsen Alouani, Mohamed Ali Mahjoub, MDAD: A multimodal and multiview in-vehicle driver action dataset, in: M. Vento, G. Percannella (Eds.), Computer Analysis of Images and Patterns, CAIP 2019, in: Lecture Notes in Computer Science, vol. 11679, Springer, Cham, 2019, pp. 518–529, http://dx.doi.org/10.1007/978-3-030-29888-3_42.
[16] N. Hautiere, A. Boubezoul, Combination of roadside and in-vehicle sensors for extensive visibility range monitoring, in: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Genova, 2009, pp. 388–393, http://dx.doi.org/10.1109/AVSS.2009.64.
[17] Hazar Chaabani, Faouzi Kamoun, Hichem Bargaoui, Fatma Outay, Ansar-Ul-Haque Yasar, A neural network approach to visibility range estimation under foggy weather conditions, Procedia Comput. Sci. 113 (2017) 466–471, http://dx.doi.org/10.1016/j.procs.2017.08.304.
[18] S.A. Taie, S. Taha, A novel secured traffic monitoring system for VANET, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, Kona, HI, 2017, pp. 176–182, http://dx.doi.org/10.1109/PERCOMW.2017.7917553.
[19] Xiangmo Zhao, Kenan Mu, Fei Hui, Christian Prehofer, A cooperative vehicle-infrastructure based urban driving environment perception method using a D-S theory-based credibility map, Optik 138 (2017) 407–415, http://dx.doi.org/10.1016/j.ijleo.2017.03.102.
[20] I. Jegham, Anouar Ben Khalifa, Pedestrian detection in poor weather conditions using Moving Camera, in: IEEE/ACS 14th International Conference on Computer Systems and Applications, AICCSA, 2017, pp. 358–362, http://dx.doi.org/10.1109/AICCSA.2017.35.
[21] J. Zhao, H. Xu, J. Wu, Y. Zheng, H. Liu, Trajectory tracking and prediction of pedestrian's crossing intention using roadside lidar, IET Intell. Transp. Syst. 13 (5) (2019) 789–795, http://dx.doi.org/10.1049/iet-its.2018.5258.
[22] G. Al-refai, M. Horani, O.A. Rawashdeh, A framework for background modeling using vehicle-to-infrastructure communication for improved candidate generation in pedestrian detection, in: IEEE International Conference on Electro/Information Technology, EIT, Rochester, MI, 2018, pp. 729–735, http://dx.doi.org/10.1109/EIT.2018.8500138.
[23] K. Chebli, Anouar Ben Khalifa, Pedestrian detection based on background compensation with block-matching algorithm, in: 2018 15th International Multi-Conference on Systems, Signals & Devices, SSD, Hammamet, 2018, pp. 497–501, http://dx.doi.org/10.1109/SSD.2018.8570499.
[24] J. Kang, I. Cohen, G. Medioni, Multi-views tracking within and across uncalibrated camera streams, in: First ACM SIGMM International Workshop on Video Surveillance, 2003, http://dx.doi.org/10.1145/982452.982456.
[25] D. Varga, T. Szirányi, A. Kiss, L. Spórás, L. Havasi, A multi-view pedestrian tracking method in an uncalibrated camera network, in: 2015 IEEE International Conference on Computer Vision Workshop, ICCVW, Santiago, 2015, pp. 184–191, http://dx.doi.org/10.1109/ICCVW.2015.33.
[26] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 267–282, http://dx.doi.org/10.1109/TPAMI.2007.1174.
[27] J. Ferryman, A. Shahrokni, PETS2009: Dataset and challenge, in: Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Snowbird, UT, 2009, pp. 1–6, http://dx.doi.org/10.1109/PETS-WINTER.2009.5399556.
[28] Y. Xu, X. Liu, Y. Liu, S. Zhu, Multi-view people tracking via hierarchical trajectory composition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, 2016, pp. 4256–4265, http://dx.doi.org/10.1109/CVPR.2016.461.
[29] T. Chavdarova, F. Fleuret, Deep multi-camera people detection, in: 16th IEEE International Conference on Machine Learning and Applications, ICMLA, Cancun, 2017, pp. 848–853, http://dx.doi.org/10.1109/ICMLA.2017.00-50.
[30] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 17–35.
[31] Masahiro Hirano, Akihito Noda, Masatoshi Ishikawa, Yuji Yamakawa, Networked high-speed vision for evasive maneuver assist, ICT Express 3 (4) (2017) 178–182, http://dx.doi.org/10.1016/j.icte.2017.11.008.
[32] T. Chavdarova, et al., WILDTRACK: A multi-camera HD dataset for dense unscripted pedestrian detection, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 5030–5039, http://dx.doi.org/10.1109/CVPR.2018.00528.
[33] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, P. Carballeira, Semantic driven multi-camera pedestrian detection, 2018, arXiv preprint arXiv:1812.10779.
[34] A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1–8, http://dx.doi.org/10.1109/CVPR.2008.4587581.
[35] X. Alameda-Pineda, et al., SALSA: A novel dataset for multimodal group behavior analysis, IEEE Trans. Pattern Anal. Mach. Intell. 38 (8) (2016) 1707–1720, http://dx.doi.org/10.1109/TPAMI.2015.2496269.
[36] C.G. Keller, M. Enzweiler, D.M. Gavrila, A new benchmark for stereo-based pedestrian detection, in: 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, 2011, pp. 691–696, http://dx.doi.org/10.1109/IVS.2011.5940480.
[37] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354–3361, http://dx.doi.org/10.1109/CVPR.2012.6248074.
[38] S. Hwang, J. Park, N. Kim, Y. Choi, I.S. Kweon, Multispectral pedestrian detection: Benchmark dataset and baseline, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Boston, MA, 2015, pp. 1037–1045, http://dx.doi.org/10.1109/CVPR.2015.7298706.
[39] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, A. López, Pedestrian detection at day/night time with visible and fir cameras: A comparison, Sensors 16 (6) (2016) 820, http://dx.doi.org/10.3390/s16060820.
[40] P. Baqué, F. Fleuret, P. Fua, Deep occlusion reasoning for multi-camera multi-target detection, in: 2017 IEEE International Conference on Computer Vision, ICCV, Venice, 2017, pp. 271–279, http://dx.doi.org/10.1109/ICCV.2017.38.
[41] Jingjing Liu, Shaoting Zhang, Multispectral deep neural networks for pedestrian detection, in: Richard C. Wilson, Edwin R. Hancock, William A.P. Smith (Eds.), Proceedings of the British Machine Vision Conference, BMVC, BMVA Press, 2016, pp. 73.1–73.13, http://dx.doi.org/10.5244/C.30.73.
[42] Yanpeng Cao, Dayan Guan, Weilin Huang, Jiangxin Yang, Yanlong Cao, Yu Qiao, Pedestrian detection with unsupervised multispectral feature learning using deep neural networks, Inf. Fusion 46 (2019) 206–217, http://dx.doi.org/10.1016/j.inffus.2018.06.005.
[43] D. König, M. Adam, C. Jarvers, G. Layher, H. Neumann, M. Teutsch, Fully convolutional region proposal networks for multispectral person detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Honolulu, HI, 2017, pp. 243–250, http://dx.doi.org/10.1109/CVPRW.2017.36.
[44] Dayan Guan, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, Michael Ying Yang, Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection, Inf. Fusion 50 (2019) 148–157, http://dx.doi.org/10.1016/j.inffus.2018.11.017.

[45] Dayan Guan, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, Christel-Loic Tisse, Exploiting fusion architectures for multispectral pedestrian detection and segmentation, Appl. Opt. 57 (2018) D108–D116, http://dx.doi.org/10.1364/AO.57.00D108.
[46] Y. Chen, H. Xie, H. Shin, Multi-layer fusion techniques using a CNN for multispectral pedestrian detection, IET Comput. Vis. 12 (8) (2018) 1179–1187, http://dx.doi.org/10.1049/iet-cvi.2018.5315, 12.
[47] Z. Zhang, W. Tao, Pedestrian detection in binocular stereo sequence based on appearance consistency, IEEE Trans. Circuits Syst. Video Technol. 26 (9) (2016) 1772–1785, http://dx.doi.org/10.1109/TCSVT.2015.2475855.
[48] Wafa Lejmi, Anouar Ben Khalifa, Mohamed Ali Mahjoub, Challenges and methods of violence detection in surveillance video: A survey, in: M. Vento, G. Percannella (Eds.), Computer Analysis of Images and Patterns, CAIP 2019, in: Lecture Notes in Computer Science, vol. 11679, Springer, Cham, 2019, pp. 62–73, http://dx.doi.org/10.1007/978-3-030-29891-3_6.
[49] Wafa Lejmi, Mohamed Ali Mahjoub, Anouar Ben Khalifa, Event detection in video sequences: Challenges and perspectives, in: 13th IEEE International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD, 2017, pp. 682–690, http://dx.doi.org/10.1109/FSKD.2017.8393354.
[50] Mehran Yazdi, Thierry Bouwmans, New trends on moving object detection in video images captured by a moving camera: A survey, Comp. Sci. Rev. 28 (2018) 157–177, http://dx.doi.org/10.1016/j.cosrev.2018.03.001.
[51] W. Ge, R.T. Collins, Crowd detection with a multiview sampler, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision – ECCV 2010, ECCV 2010, in: Lecture Notes in Computer Science, vol. 6315, Springer, Berlin, Heidelberg, 2010, http://dx.doi.org/10.1007/978-3-642-15555-0_24.
[52] Kihong Park, Seungryong Kim, Kwanghoon Sohn, Unified multi-spectral pedestrian detection based on probabilistic fusion networks, Pattern Recognit. (2018) http://dx.doi.org/10.1016/j.patcog.2018.03.007.
[53] R. Panda, A.K. Roy-Chowdhury, Multi-view surveillance video summarization via joint embedding and sparse optimization, IEEE Trans. Multimed. 19 (9) (2017) 2010–2021, http://dx.doi.org/10.1109/TMM.2017.2708981.
[54] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards realtime object detection with region proposal networks, in: Advances in Neural Information Processing Systems, NIPS, 2015, http://dx.doi.org/10.1109/TPAMI.2016.2577031.
[55] J. Huang, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, 2017, 3296–3297, http://dx.doi.org/10.1109/CVPR.2017.351.
[56] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, CVPR'05, San Diego, CA, USA, 2005, pp. 886–893, http://dx.doi.org/10.1109/CVPR.2005.177.
[57] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987, http://dx.doi.org/10.1109/TPAMI.2002.1017623.
[58] T. Matsukawa, T. Okabe, E. Suzuki, Y. Sato, Hierarchical Gaussian descriptor for person re-identification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1363–1372, http://dx.doi.org/10.1109/CVPR.2016.152.
[59] R. Kasturi, et al., Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 319–336, http://dx.doi.org/10.1109/TPAMI.2008.57.
[60] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 743–761, http://dx.doi.org/10.1109/TPAMI.2011.155.
[61] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015, pp. 1–14.

Dr. Anouar Ben Khalifa : received the engineering degree (2005) from the National Engineering School of Monastir – Monastir university (Tunisia), a M.Sc. degree (2007) and a Ph.D. degree (2014) in Electrical Engineering, Signal Processing, System Analysis and Pattern Recognition from the National Engineering School of Tunis – Tunis ElManar university (Tunisia). He is now Associate Professor in Electrical and Computer Engineering at the National Engineering School of Sousse–Sousse university (Tunisia). He is a Founding member of the LATIS research labs (Laboratory of Advanced Technology and Intelligent Systems). He is the head of the Department of Industrial Electronic Engineering at the National Engineering School of Sousse (From 2016 to 2019). His research interests are Artificial Intelligence, Pattern Recognition, Image Processing, Machine Learning and Information Fusion. anouar.benkhalifa@eniso.rnu.tn

Dr. Ihsen Alouani: is an Associate Professor at the IEMN-DOAE lab in the Polytechnic University Hauts-De-France, France. He got his PhD from the Polytechnic University Hauts-De-France, a MSc and engineering degree from the National Engineering School Sousse, Tunisia. He is the head of "Cyber-defense and Information Security" Master's program. His research focus is on Intelligent Transportation Systems, Hardware acceleration and security. ihsen.alouani@uphf.fr

Pr. Mohamed Ali Mahjoub: is Professor at National Engineering School of Sousse (university of Sousse – Tunisia) and member of research LATIS laboratory, team signals, image and document. He received the MSc in computer science in 1990, and Phd and HDR in electrical engineering, signal processing and system analysis, from the National School of Engineers of Tunis, Tunisia, in 1999 and 2013 respectively. His research interests include dynamic bayesian network, computer vision, pattern recognition, HMM, and data retrieval. His main papers have been published in international journals and conferences. mohamedali.mahjoub@eniso.rnu.tn

Pr. Atika Rivenq: is a Full Professor at the Department of Electronic Engineering, IEMN Lab, University of Valenciennes (UPHF), France. She is responsible of ComNum Group at IEMN. She was graduated Engineer from the ENSIMEV engineering school in 1993, received Diploma of the M.S. degree in electronic engineering in 1993 and then her Ph.D. degree in 1996 from the University of Valenciennes (France). She is responsible of SYFRA platform (Systems for smart road applications) with the IEMN-DOAE Lab. Main activity is in digital communications applied to intelligent transports systems and security: V2X communications (4G/5G, UWB, ITS-G5, Full Duplex), Cyber security and Advanced Perception (Radar, UWB, Detection of vulnerable persons, Deep learning). She Participates to many national and European projects dedicated to C-ITS and inter vehicles communications especially using ITS-G5 and Cellular systems. tika.menhaj@uphf.fr