Machine Vision and Applications (2021) 32:109
https://doi.org/10.1007/s00138-021-01231-4

ORIGINAL PAPER

On the safety of vulnerable road users by cyclist detection and tracking

M. García-Venegas[1] · D. A. Mercado-Ravell[2] · L. A. Pinedo-Sánchez[1] · C. A. Carballo-Monsivais[1]

Received: 8 October 2020 / Revised: 27 April 2021 / Accepted: 15 July 2021 / Published online: 12 August 2021

Abstract
Timely detection of vulnerable road users is of great relevance to avoid accidents in the context of intelligent transportation systems. In this work, detection and tracking is acknowledged for a particularly vulnerable class of road users, the cyclists. We present a performance comparison between the main deep learning-based algorithms reported in the literature for object detection, such as SSD, Faster R-CNN and R-FCN along with InceptionV2, ResNet50, ResNet101, Mobilenet V2 feature extractors. In order to identify the cyclist heading and predict its intentions, we propose a multi-class detection with eight classes according to orientations. To do so, we introduce a new dataset called "CIMAT-Cyclist", containing 20,229 cyclist instances over 11,103 images, labeled based on the cyclist's orientation. To improve the performance in cyclists' detection, the Kalman filter is used for tracking, coupled together with the Kuhn–Munkres algorithm for multi-target association. Finally, the vulnerability of the cyclists is evaluated for each instance in the field of view, taking into account their proximity and predicted intentions according to their heading angle, and a risk level is assigned to each cyclist. Experimental results validate the proposed strategy in real scenarios, showing good performance.

Keywords VRUs · Deep learning · CNN · Detection and tracking · Kalman filter

1 Introduction

In recent years, significant progress has been achieved in the care and protection of vulnerable road users (VRUs), including pedestrians, motorcyclists and cyclists. This efforts began with the creation of road laws and rules that attempt to convert roads into safer spaces for these users [52]. On the other hand, several researches have been done working on vision-based detection systems for VRUs, specially pedestrians, along with improvements in the field of Intelligent Transportation Systems (ITSs) for traffic monitoring and the development of Advanced Driver-Assistance Systems (ADASs), such as collision avoidance.

However, the number of accidents and deaths on the roads continues to climb at high rate. As evidenced by the World Health Organization (WHO), there have been 1.35 million annual deaths and up to 50 million injuries are reported each year, becoming the eighth cause of death in the world, where more than half of all this road deaths were VRUs, mainly pedestrians, cyclists and motorcyclists. In addition, unfortunately, one cyclist is dying every 12.36 min on the world's roads every day [53].

Provided that the most affected VRUs are the pedestrians, research on the detection and monitoring of pedestrians have received most of the attention [6,15,21,28]. Unfortunately, little attention has been paid to cyclists, even though the lack of special infrastructure, adequate protection and road safety culture makes them particularly vulnerable to road accidents. Furthermore, in contrast to pedestrian's detection, the cyclist's detection task presents other challenges, mainly due to the cyclists' visual complexity, variety of possible orientations; aspect ratios, pose and appearance, along with the lack of labeled datasets [26] and the presence of occlusions and cluttered backgrounds [29,30,58].

Former techniques for detection of VRUs included classic artificial vision approaches for pedestrian detection, which were mainly implemented using Histogram of Gradients Oriented (HOG) for feature extraction and Support Vector Machine (SVM) for classification [9,59]. The HOG-SVM combination proved to be a good option for human and bicy-

✉ D. A. Mercado-Ravell
diego.mercado@cimat.mx

M. García-Venegas
marichelo.garciaV@gmail.com

[1] Center for Research in Mathematics CIMAT AC, Campus Zacatecas, Zacatecas, Mexico

[2] Cátedras CONACyT, Center for Research in Mathematics CIMAT AC, Campus Zacatecas, Zacatecas, Mexico

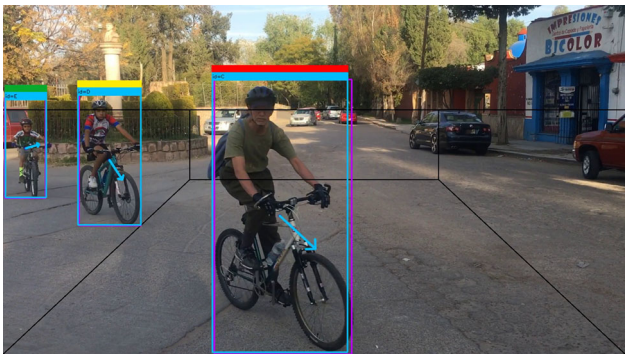

Fig. 1 Monitoring the safety of the cyclists based on their proximity, along with their predicted intentions of motion depending on their orientation (color figure online)

cle detection, until the arrival of deep learning (DL)-based algorithms within the last decade. With the boom of Convolutional Neuronal Networks (CNNs), more recent works like [6,15,28,46] applied them to a region proposal, with significant improvements in precision. With the rapid development of DL, more powerful machine learning techniques have emerged, overcoming the problems existing in traditional architectures. Now, with the CNNs, it is possible to learn semantic, high-level, deeper features [60]. This converts them in a powerful tool for detection and classification, specially when further combined with the great advances in hardware for parallel processing, with the recent developments in Graphics Processing Units (GPUs).

In this work, we are interested in the road safety of a particular kind of VRU whose dynamics obey nonholonomic constrains. Henceforth, we propose a multi-class detection strategy based on the cyclist orientation. Accordingly, we introduced a new dataset called "CIMAT-Cyclist", containing 20,229 cyclist instances over 11,103 images, which has been labeled based on the cyclists orientation.

In order to accomplish cyclists' detection, we make use of the state-of-the-art DL techniques, such as, Single Shot MultiBox Detector (SSD), Faster Region-based Convolutional Network (Faster R-CNN) and Region-based Fully Convolutional Networks (R-FCN) meta-architectures in combination with MobilenetV2, InceptionV2, Residual Network with 50-layers (ResNet50), Residual Network with 101-layers (ResNet101) and InceptionResNetV2 feature extractors. Taking advantage of Transfer Learning with pre-trained models and TensorFlow Object Detection API (Application Programming Interface), we implemented different models for cyclist's detection and evaluate them thoroughly. Experimental results suggested that Faster R-CNN with InceptionV2 offers the best alternative for the cyclists detection and orientation detection task when greater precision is required; however, when a better time response is needed, the best option is given by SSD with MobilenetV2.

Finally, we propose a complete strategy to monitor the safety of the VRUs in the context of ITS. The strategy consists in detecting the VRUs and their orientations at high rate using SSD with MobilenetV2; then, a Kalman filter (KF) is implemented along with the Kuhn–Munkres algorithm for tracking and data association along time. Afterward, a risk evaluation is preformed based on the VRUs' proximity, along with their predicted intentions of motion based on their orientation, as shown in Fig. 1. Notice that given the nonholonomic nature of the cyclists dynamics, knowledge of their orientation provides a good notion on their movement intentions. At the end, a risk level is assigned to each VRU in real time and notified to the driver in case of danger.

The main contributions of this paper are then summarized as follows:

- A strategy for the risk evaluation of the VRU, based on detection and tracking using SSD with Mobilenet V2 in combination with the Kalman filter and the Kuhn–Munkres algorithm. Furthermore, it integrates an alternative technique for the detection of the cyclist's orientation, which allows for a good notion of the cyclist's intention of movement, and makes it possible to predict future risks.
- Creation of a new dataset which has been labeled based on the cyclist's orientation, containing 20,229 cyclist instances over 11,103 images.
- A more in-depth and updated evaluation of state-of-the-art techniques to perform this cyclist detection task. Particularly to determine a good trade-off between precision and time response under different scenarios.

The remaining of the paper is organized as follows: Sect. 2 presents related works about cyclist's detection and tracking. Afterward, Sect. 3 introduces a new cyclist detection dataset. Then, Sect. 4 describes the methodology for cyclist and orientations detection using CNNs, along with a validation study. Meanwhile, Sect. 5 presents the complete strategy for risk evaluation of VRUs using detection and tracking. Also, Sect. 6 studies the performance of the strategy in real scenarios. Finally, Sect. 7 discusses the conclusions and future work.

2 Related works

This section presents the most relevant works regarding cyclists' detection and tracking.

2.1 Cyclist detection

Up to now, a lot of improved models have appeared for generic object detection, including Fast R-CNN that jointly

optimizes classification and bounding box regression; Faster R-CNN which introduces an additional Region Proposal Network (RPN) that can predict bounding box and score at each position simultaneously; and SSD that accomplishes object detection via regression. All of them have implemented important improvements in accuracy and execution time, and can even be used for real-time applications. In this scenario, SSD presents an interesting solution, providing the fastest detection at the cost of some precision. Then, at current state of the art, there is a significant trade-off between precision and time response, and the best detector is to be chosen according to the application.

Accordingly, CNN methods have been used for cyclist detection, being Fast R-CNN [30,51], Faster R-CNN [6,43] and YOLO [32,45] the most studied ones. For example, in [51] a unified joint detection framework for pedestrians and cyclists was presented based on Fast R-CNN to estimate three categories: pedestrians, cyclists and background, using the target candidate region selection method (MIOP) along with VGG8, VGG11 and VGG16 feature extractors. It was trained with the Tsinghua-Daimler Cyclist Benchmark dataset (TDCB) presented in [29]. This dataset has also been used in [32] where authors proposed Aggregated Channel Feature-Region Proposal-YOLO (ACF-PR-YOLO) for cyclist detection. Meanwhile, in [43] Faster R-CNN was used for detecting instances of cyclists in depth images, but it requires data from an extra sensor such as the laser scanner. In addition, Chen et al. [6] evaluated the pedestrian and cyclist detection using thermal images. Also, in most recent investigations, [45] studied cyclist detection using the Tiny YOLO v2 algorithm with a dataset proposed on [29], also in [1] Yolov1 has been used on Pascal VOC datasets 2007/2012 for bike detection. In [29], a new method called Stereo-Proposal-based Fast R-CNN (SP-FRCN) was introduced to detect cyclists using their own dataset TDCB, which contains VRUs including pedestrians, cyclists and motorcycles instances, recorded from a moving vehicle in the urban traffic of Beijing. It divides the cyclist samples into three classes: narrow, intermediate and wide, based on the aspect ratio of bikes. In a similar fashion, in [30] the same authors presented another unified framework for concurrent pedestrian and cyclists detection, including a proposal method called Upper Body-Multiple Potential Regions (UB-MPR) for generating object candidates and using Fast R-CNN for classification and localization.

As can be seen, most of these studies have focused only in the cyclist detection task. Nevertheless, in the context of ITS and VRUs' safety, detecting the objects of interest is not enough, provided that VRUs are constantly moving and changing their appearance. It is then of great interest to further gather information about the movement of the cyclists, and try to predict their intentions in the near future to determine on-time whether or not they will be in danger. While pedestrians may be unpredictable in the direction of their movement, for the particular case of nonholonomic vehicles, such as bicycles, which always move forward in normal conditions, it is of great interest to also know their orientation, since it provides great insight about their future movement, which is crucial to prevent accidents.

Regardless of the importance of detecting both the position and orientation to prevent accidents, orientation detection is rarely considered in cyclist's detection [50]. Tian et al. [50] proposed dividing the cyclists into eight subcategories based on orientation using the KITTI dataset [13], analogously to an idea previously used for vehicle detection [36]. For each orientation a detector is built in a cascaded structure, using classical approaches. Besides, they used a geometric method for Region of Interest (RoI) extraction and Kalman filters to estimate cyclists' trajectories, with a total of 16 detectors. Nevertheless, the use of multiple detectors and traditional techniques significantly compromises precision and increases the amount of calculations required, when compared to modern CNN-based techniques.

In the growing interest of protecting VRUs and not only knowing their location, but also their orientation, the KITTI dataset has become a benchmark for the current work [5,17,19]. This dataset provides 3D bounding box annotations, for object classes such as cars, vans, trucks, pedestrians, cyclists and trams, and it is evaluated in three regimes: easy, moderate and hard, depending on the levels of occlusion and truncation. For example, Guindel et al. [18] proposed a joint detection and viewpoint estimation system with a monocular camera using Faster R-CNN meta-architecture with VGG16 feature extractor, for determining the orientation of the three objects: car, pedestrian and cyclist. For estimation of the object's viewpoint, they adopt discrete pose estimators to partition the view sphere into a predefined number of bins, and compute the viewpoint as the weighted average of adjacent viewpoint bin centers, using their respective estimated probabilities provided by the network. In a posterior work in [19], the same authors proposed an approach for recognition and 3D localization of dynamic objects on images from a stereo camera, with a stereo-based 3D reconstruction of the environment, using the KITTI dataset with 1626 samples of cyclist; finally, they implemented their system on an intelligent vehicle.

As can be observed, nowadays it is not only sufficient to carry out object detection, but it is also important to include their orientation. In this case, the trend of the most recent works is the use of 3D stereo vision and continues to take advantage of the KITTI dataset. Lamentably, this dataset only provides a small amount of cyclist instances (no more than 2000 [19,29]). From our part, we propose a monocular vision-based approach with a new dataset specialized in cyclists that contains over 20,229 instances labeled according to the

orientations and used for training a multi-class orientation detector.

2.2 Cyclist tracking

Although modern CNN-based techniques have significantly improved the object detection task, they are not exempt to failure over time, which in the context of VRUs safety may translate in fatal accidents. Consequently, the implementation of tracking techniques is still of great help in order to increase the robustness against false negative detections and object occlusions, as well as to improve the consistency over time and even predict the future movement of the target, hence augmenting the VRUs safety.

In this sense, cyclist's tracking has been addressed in some works, mainly using Kalman filters (KF) and extended Kalman filters (EKF); for example, Cho et al. [7] performed the detection and tracking of bicycles using Latent SVM for training the detector, and EKF for tracking the position and velocity of the bicycle in the vehicle's coordinates. Also, Jung et al. [25] made three classifiers to consider the direction of the bicycle: front, left and right. HOG was used for feature extraction, along with Real Adaboost classification. Tracking was performed using a particle filter (PF). Meanwhile, Tian et al. [49] implemented the KF together with an optical filter (feature points) to estimate the cyclist's trajectory, using HOG cascade detectors. The authors used a constant velocity model on the bounding box coordinates, and a Probabilistic Data Association (IPDA) algorithm.

More recently, Zhang et al. [58] proposed a method for cyclist detection and tracking based on multi-layer laser scanner. For cyclist tracking, Multiple Hypothesis Tracking (MHT) algorithm and Kalman filter based on Current Statistical (CS) were used. Meanwhile, Maurya et al. [35] applied a method based on DL for the detection of pose in 2D using Part Affinity Fields (PAF) to detect people and cyclists, as well as tracking them, in addition to measuring the degree of vulnerability as a function of the height of the object's skeleton and the division of the image into regions to alert the driver.

In our case, given its faster response time, we propose to use SSD with MobilenetV2 for cyclists' detection at high rate, coupled with KF for tracking in real time, using a constant velocity model on the bounding boxes obtained from the detector. Besides, a second SSD with MobilenetV2 model for multi-class detection is used to estimate the cyclists heading angle. In addition, for the association of matching detections and trackers, we use the Kuhn–Munkres algorithm [24]. Finally, the image is partitioned into 5 risk regions, and a level of risk (danger, warning or safe) is assigned to each instance in accordance with its location in the risk regions, and their predicted intentions to cross to a different region based on their orientation.

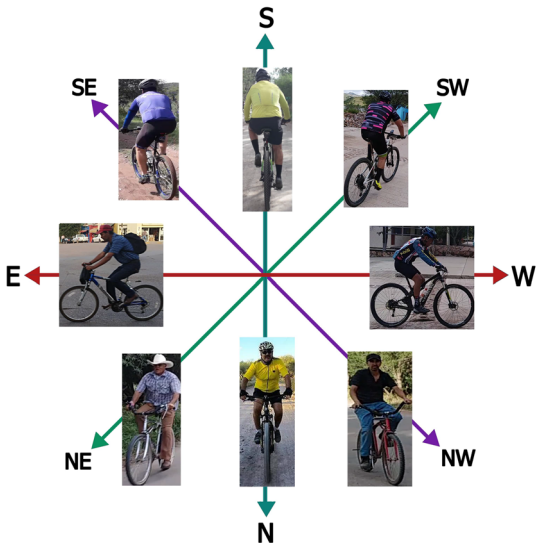

Fig. 2 Cyclists instances are labeled into 8 classes according to orientation: CyclistN, CyclistNE, cyclistE, cyclistSE, cyclistS, cyclistSW, cyclistW and cyclistNW

3 Cyclist image dataset

The dataset is essential for the proper training of object detectors. As we know, the first step in building an object detector is the preparation of a dataset with labeled images. In the case of public datasets available with cyclist's instances, only two are to be found: TDCB [29], which has been mainly considered for the detection of pedestrians and cyclists; however, it is not labeled based on the orientation of the cyclist, and the KITTI dataset [13], which unfortunately provides a very limited number of cyclist's instances (less than 2000).

For this reason, we provide a new dataset, called "CIMAT-Cyclist", with 20,229 instances over 11,103 images, labeled according to eight different classes of orientation, as observed in Fig. 2. It was collected combining images available on the Internet with images taken from our surroundings, hence improving the generalization capacity of the algorithms in the cyclists' detection. Furthermore, we believe that the new labels provided according to the cyclist orientation, may be helpful to determine the cyclist heading and predict their movement, which is crucial to prevent accidents on the roads.

Among the 11,103 images, roughly 60% was collected from sport events and urban areas in the center of Mexico, while the remaining 40% was obtained from websites such as pixabay, pexels, freephotos, and others. CIMAT-Cyclist allows to complement the existing datasets for the detection of cyclists, besides being useful to evaluate the capacity of the state-of-the-art models for the detection of cyclists and their orientation. In both cases, 80% of the images were used for the training set and 20% for the test set.

According to the type of labels used, we distinguish two versions of our dataset, namely CIMAT-CyclistV1 and

Table 1 CIMAT-CyclistV2 dataset for cyclists' orientation

Class orientation	Total	Training	Test
CyclistN	3870	3100	770
CyclistNE	3023	2406	617
CyclistE	2232	1789	443
CyclistSE	1849	1513	336
CyclistS	2427	1931	496
CyclistSW	1864	1478	386
CyclistW	1918	1544	374
CyclistNW	3046	2438	608
Total	20,229	16,199	4,030

At current time we provide 20,229 instances over 11,103 images, where 80% of the images were used for the training set and 20% for the test set. By now, we focus mainly in large size instances

CIMAT-CyclistV2. For dataset CIMAT-CyclistV1, used for single class detection of cyclists, the labels consist in bounding boxes enclosing every instance. Meanwhile, in dataset CIMAT-CyclistV2, all cyclists on the images were labeled using bounding boxes, and divided into eight classes according to their orientation, in order to identify the direction of movement of the cyclist. As suggested in [18], eight categories are a good choice to represent the cyclist's orientation. These classes are named inspired by the compass rose as Cyclist N, Cyclist NE, Cyclist E, Cyclist E, Cyclist SE, Cyclist S, Cyclist SW, Cyclist W and Cyclist NW, as depicted in Fig. 2. Meanwhile, Table 1 shows the number of each class instances available in the dataset.

Now, the new generated dataset will be used to train and evaluate some of the most relevant CNN-based methods for both tasks, the cyclist detection and the multi-class orientation detection.

4 Cyclist detection

Deep learning and more in particular CNN-based methods are considered to be the best for object detection up to today [15,60]. Hence, in this work we use them for the cyclist detection task. In order to establish a baseline comparison between the main object detection algorithms for this particular task, we have studied three of the main meta-architectures reported in the literature, such as Faster-RCNN [40], R-FCN [8], and SSD [34]. In this section, we present a thoroughly analysis to identify and validate the better suited models for both, the cyclist detection and multi-class orientation detection, to be used with the proposed VRUs risk evaluation method. Figure 3 summarizes the overall methodology employed in this work.

For generic object detection, two types of frameworks have been introduced in the literature, "region proposal based" and "regression/classification based" [54,60]. The former pursues the traditional object detection pipeline, generating region proposals at first and then classifying each proposal into different object categories. These methods include R-CNNs [15], Fast R-CNN [14], Faster R-CNN [40] and Region-based Fully Convolutional Network (R-FCN) [8]. The second framework considers object detection as a regression or classification problem, implementing a unified framework to accomplish the final result, which involves categories and locations. These methods include Single Shot MultiBox Detector (SSD) [34], Multibox [10], You Only Look Once (YOLO) [38], YOLOv2 [39], among others. In general, "region proposal-based" methods are known to be more accurate, while "regression/classification-based" algorithms are significantly faster [60]. Faster-RCNN [40], R-FCN [8], and SSD [34] have been selected for the study.

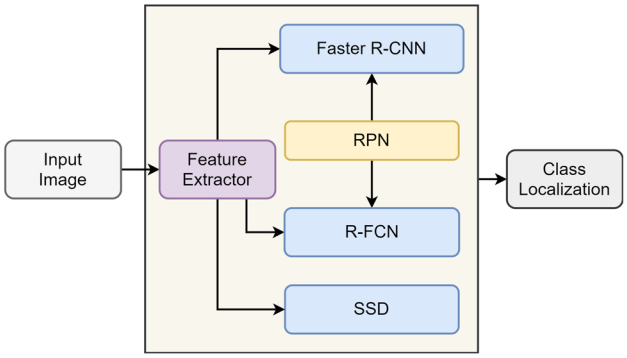

Fig. 3 Meta-architectures' overall strategy. First an input image is needed. As part of the meta-architecture, one of the feature extractors is chosen: MobileNetV2, InceptionV2, ResNet50, ResNet101 or InceptionResNetV2. Then, we can notice that both Faster R-CNN and R-FCN use RPN to generate object proposals and are well known for their superior precision. On the other hand, SSD uses multi-scale feature maps for detection in a single stage, considerably reducing the execution time

In all meta-architectures, first, the images are processed by a feature extractor to obtain high level features. The choice of the feature extractor is very important, since the number of parameters and types of layers directly affect memory usage, time response, complexity and performance of the detector [22]. In this paper, five state-of-the-art feature extractors are considered: MobilenetV2 [44], InceptionV2 [48], ResNet50 [20], ResNet101 [20] and InceptionResNetV2 [47], provided that they have been efficient for the task of object detection.

We propose cyclists detection estimation using the framework available on [22], which consist of a single convolutional network, trained with a mixed regression and classification objective, and use sliding window style predictions.

Seven models were considered using a meta-architecture combined with some feature extractor, as shown in Table 2. These models were trained for the detection of cyclists using

Table 2 Seven different models were generated from the combination of a meta-architecture and a feature extractor

Models	
Meta-architecture	Feature extractor
SSD	MobilenetV2
	InceptionV2
RFCN	ResNet101
Faster RCNN	ResNet50
	InceptionV2
	ResNet101
	InceptionResNetV2

the label: "cyclist", and the same seven models were also studied for the multi-class detection of cyclist orientation, using the 8 classes already defined in the dataset according to orientation.

4.1 CNN validation

To perform the experiments, our dataset has been divided into an 80% training set and 20% testing set. In order to measure the speed of the detector, one video of 1920 × 1080 pixels with 435 frames taken in the streets has been used for obtaining the FPS required by each model using our current hardware. The implementation has been carried out in a computer using Windows 10 64-bit Operative System, with a processor Intel® Core™ i7-9750H and a dedicated GPU NVIDIA® GeForce® RTX 2070 (8GB GDDR6). For the neural network implementation, we make use of the TensorFlow-gpu V1.14.0 API, along with CUDA v10.0. Finally, the evaluation has been carried out with the help of the package python *pycocotools*.

For evaluation, we have used the COCO detection Metrics [31] and Open Images V2 detection metrics, available on [22], for the comparison of each meta-architecture. COCO metrics have been selected mainly because COCO-trained models were employed by means of transfer learning, while Open Images V2 metrics have allowed to evaluate multi-class cyclist orientation.

In COCO, 12 metrics are handled for describing the performance of an object detector, and all of them were computed in the present study, but only the most representative ones are presented.

In order to evaluate each meta-architecture using the CIMAT-Cyclist dataset, the performance is calculated in terms of average precision (AP), which is introduced in the Pascal VOC Challenge [11].

The AP summarizes the shape of the *precision/recall* curve and is defined as the mean precision at a set of eleven equally spaced recall levels [0, 0.1, . . . , 1]

$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \left(P_{\mathrm{interp}}(r) \right) \tag{1}$$

where $P_{\mathrm{interp}}(r)$ is an interpolation function that takes the maximum measured precision at each recall level [11]. For single-class detection, there is no distinction between average precision (AP) and mean AP (mAP).

Another important metric is the intersection over union (IoU), which is used to obtain the area of overlap between the predicted bounding box $\mathrm{BB_p}$ and the ground truth bounding box $\mathrm{BB_{gt}}$

$$\mathrm{IoU} = \frac{\mathrm{area}\left(\mathrm{BB_p} \cap \mathrm{BB_{gt}}\right)}{\mathrm{area}\left(\mathrm{BB_p} \cup \mathrm{BB_{gt}}\right)} \tag{2}$$

Then, AP can be also averaged over multiple IoU values between 0.5 and 0.95 thresholds, such as AP@.50IoU (PASCAL VOC metric) and AP@.75IoU. Other scores are average recall (AR), which measures the maximum recall given a fixed number (1, 10 or 100) of detections allowed in the image. Both AP and AR are averaged over three instance sizes: small, medium and large.

Also, as an important part of the functioning of each meta-architecture, loss functions are evaluated to help minimize the error in classification (cls) and localization (loc) of an object of interest. Each training RoI is labeled with a ground-truth class u and a ground-truth bounding-box regression target v. A multitasking loss L in each labeled RoI is used to jointly train the classification and regression of the bounding box, given by [14,40] :

$$L\left(p, u, t^u, v\right) = L_{\mathrm{cls}}\left(p, u\right) + \lambda[u \geq 1] L_{\mathrm{loc}}\left(t^u, v\right) \tag{3}$$

where p is predicted class scores, u is the true class scores, t^u is the predicted box coordinates, while the function of the Iverson bracketed indicator $\lambda[u \geq 1]$ evaluates to 1 when $[u \geq 1]$ and 0 otherwise, having as a convention that the background class is $u = 0$. L_{cls} is the classification loss, while L_{loc} represents the localization loss.

Other important aspect to evaluate the performance of the detection algorithms is the detection speed in frames per second (FPS) [33]. In this sense, we calculated the mean FPS for each detection model using the same hardware on a video of 1920 × 1080 with 435 frames.

4.2 Model evaluation

In the following, evaluation and implementation details of the selected meta-architectures and feature extractors are described. We provide a thoroughly comparison of the more

Fig. 4 Visual result examples of four representative models for single-class cyclist detection: **a** Faster R-CNN with InceptionV2, **b** Faster R-CNN with InceptionResNetV2, **c** R-FCN with ResNet101 and **d** SSD with MobilenetV2. The bounding box and score are displayed when a cyclist is detected. Most detectors showed acceptable performance. Faster R-CNN InceptionV2 managed to identify all cyclist within the image with high score. Meanwhile, R-FCN with ResNet101 generated several undesired bounding boxes with different scores for the same cyclist instance. On the other hand, SSD with MobilenetV2 did not manage to detect all the cyclist, specially the ones further away (color figure online)

relevant multiple object detection meta-architectures available on [22], using the CIMAT-Cyclist dataset. The study is conducted in two stages: first a comparison of the performance of the main models for single class object detection is provided, for the particular case of cyclist detection in Sect. 4.3. The second stage in Sect. 4.4 consists in the proposal of a new multi-class detection strategy that further takes into account the orientation of the cyclists, which we consider to be of relevance in the context of road safety of VRUs. In order to do so, we take advantage of the new dataset for cyclist detection with orientation labels CIMAT-CyclistV2.

Detection examples of the behavior of four representative models for cyclist detection that have been trained and evaluated with the CIMAT-CyclistV1 dataset are shown in Fig. 4. Similarly, visual example results for multi-class cyclist's orientation detection are shown in Fig. 5.

This work offers an updated comparison study of the state-of-the-art techniques applied to detect cyclists. We focus on identifying the best meta-architectures in terms of average precision (AP), detection speed in frames per second (FPS) and a good trade-off between both.

4.3 Single-class detection

In this subsection, we focus on the single-class cyclist detection, in order to identify which state-of-the-art techniques are the best suited for detection of these particular case of VRU.

The seven models that have been selected by combining a meta-architecture with a specific feature extractor, shown in Table 2, have been trained for the instance "Cyclist", using the dataset CIMAT-CyclistV1, and evaluated with the metrics previously explained. Even though all the metrics proposed by COCO [31] have been analyzed in this work, only the most relevant ones are presented in Table 3.

For cyclist's detection, Faster R-CNN meta-architecture using InceptionResNetV2 feature extractor was found to be the most precise, as can be appreciated in Table 3. In this table the most relevant metrics of the evaluation protocol COCO are shown for each of the models (meta-architecture and feature extractor) in this case, Faster R-CNN with InceptionResNetV2 obtained the highest value in all the metrics.

Considering the size of the objects is important for the cyclist's detection on the roads, since it considerably affects

Fig. 5 Multi-class orientation detection examples of four representative models: **a** Faster R-CNN with InceptionV2, **b** Faster R-CNN with InceptionResNetV2, **c** R-FCN with ResNet101 and **d** SSD with MobilenetV2. For each class a different colored bounding box is displayed. For this example, only Faster R-CNN with InceptionV2 and R-FCN with ResNet101 managed to detect all cyclists within the image. A problem that was identified for each model is that similar classes such as CyclistNW (magenta) and CyclistNE (gray) are hard to differentiate (color figure online)

the detection performance (further away objects are smaller and harder to detect), moreover, it is a good indicator of how close cyclists are to the camera, where larger objects present higher collision risks. We present in Table 3, AP_l and AR_l metrics for large size objects, while AP_m and AR_m stand for medium size instances. It can be observed how their performance changes with respect to the size of the cyclist instances, Faster meta-architecture R-CNN and R-FCN were the more efficient if we consider the medium and small instances as opposed to SSD.

The obtained results are consistent with the ones reported in the literature for object detection [23,37], also supporting the quality of the new dataset. Nevertheless, it is important to notice that there is a trade-off between precision and detection time, where the most precise algorithm FasterRCNN with InceptionResNetV2 is up to 30 times slower than the fastest model, SSD with MobilenetV2, when running on our baseline hardware.

In addition, when evaluating time response, Faster R-CNN and R-FCN meta-architectures are strongly overcome by the SSD meta-architecture, presenting competitive precision for large size instances. This is why it is important to select the best model according to the application, and determine good trade-offs between the parameters of interest. In particular, for the case of VRU safety on the road with ITS, it is important to detect them correctly, but it is also relevant to detect them on time. Furthermore, having detections at high rate can be of great use, specially when combined with other algorithms, for example to track the objects in real time. As is the case with this study, SSD-MobilenetV2 has been chosen as the detection model along with the implementation of the Kalman filter for cyclist tracking.

4.4 Multi-class detection

Cyclist detection is not a trivial task; however, nowadays it has become a much more challenging problem when considering the cyclist's orientation, reason why we aim to also identify the cyclist's direction of movement based on his orientation.

In this case we have provided and used for training the CIMAT-CyclistV2 dataset, with the cyclist's orientation

Table 3 COCO metrics for each model for cyclist's detection, AP@.5IoU, AP@.75IoU, AP and AR large and medium sizes of cyclist's instances, classification loss (Cls loss), localization loss (Loc loss) and FPS

AP	AP@.5	AP@.75	AP_l	AP_m	AR_l	AR_m	Cls loss	Loc loss	FPS	Architecture	Extractor
0.573	0.868	0.633	0.622	0.165	0.689	0.307	2.562	0.457	46.826	SSD	InceptionV2
0.639	0.923	0.737	0.691	0.201	0.741	0.331	1.816	0.292	**57.864**	SSD	MobilenetV2
0.754	0.972	0.889	0.787	0.440	0.821	0.592	0.065	0.039	8.630	RFCN	ResNet101
0.772	0.978	0.893	0.806	0.455	0.836	0.592	0.057	0.031	12.392	FasterRCNN	InceptionV2
0.787	0.980	0.903	0.819	0.470	0.848	0.616	0.048	0.030	6.572	FasterRCNN	ResNet50
0.803	0.982	0.910	0.836	0.493	0.864	0.619	0.041	0.026	5.689	FasterRCNN	ResNet101
0.819	**0.983**	**0.939**	**0.847**	**0.543**	**0.876**	**0.668**	0.046	0.026	1.470	**FasterRCNN**	**Inc.ResNetV2**

All models exceed 86% AP@.5IoU. Faster R-CNN meta-architecture with InceptionResNetV2 feature extractor was the most precise and SSD meta-architecture with MobilenetV2 feature extractor was the speediest, while Faster R-CNN meta-architecture with InceptionV2 offers a good trade-off between precision and time response, but SSD meta-architecture with MobilenetV2 feature extractor was the best choice for real-time applications, specially if far away objects are neglected
Bold characters aim to highlight the best result for every criterion

Table 4 COCO metrics for each model, AP@.5IoU, AP@.95IoU, AP and AR for large and medium sizes of cyclist instances, classification loss (Cls loss), localization loss (Loc loss) and FPS

AP@.5	AP@.75	mAP	AP_l	AP_m	AR_l	AR_m	Cls loss	Loc loss	FPS	Architecture	Extractor
0.577	0.460	0.396	0.429	0.062	0.722	0.257	3.089	0.304	34.007	SSD	InceptionV2
0.589	0.471	0.402	0.435	0.062	0.718	0.312	3.101	0.329	**54.710**	**SSD**	**MobilenetV2**
0.663	0.599	0.503	0.535	0.163	0.804	0.572	0.166	0.049	7.654	RFCN	ResNet101
0.664	0.616	0.526	0.557	0.199	**0.847**	**0.609**	0.130	**0.029**	1.529	FasterRCNN	Inc.ResNetV2
0.672	0.617	0.517	0.424	0.093	0.820	0.590	0.134	0.035	9.103	FasterRCNN	InceptionV2
0.693	0.635	0.544	0.576	0.195	0.838	0.612	0.136	0.030	5.327	FasterRCNN	ResNet101
0.712	**0.647**	**0.545**	**0.578**	**0.201**	0.816	0.599	**0.117**	0.035	6.099	**FasterRCNN**	**ResNet50**

Faster R-CNN meta-architecture with ResNet50 feature extractor was the most precise, while SSD meta-architecture with MobilenetV2 feature extractor was the speediest
Bold characters aim to highlight the best result for every criterion

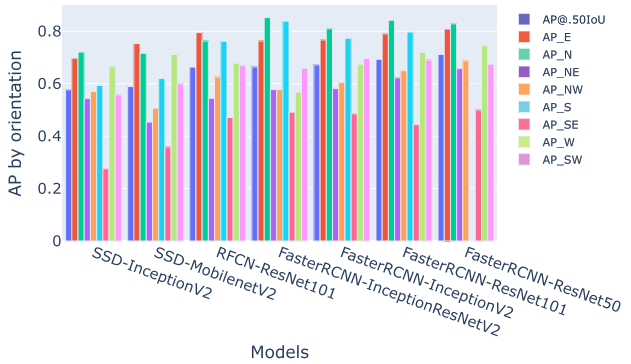

Fig. 6 Average precision with threshold 0.5 on IoU for each class. Faster R-CNN with ResNet50 was the most consistent meta-architecture for all classes. In all the models it is observed that the number of instances by class considerably affects the detection performance (color figure online)

labels, and evaluated with the metrics explained in Sect. 4. Analogous to the previous subsection, the most relevant metrics are presented in Table 4. Also, since the orientation detection problem is accomplished as a multi-class detection, with eight different classes, we further employ the Open Image V2 metrics such as AP by category of cyclist orientation on AP@.50IoU, as depicted in Fig. 6 along with Table 5.

It is noteworthy to point out that the new introduced labeled dataset CIMAT-CyclistV2 is still under construction, and the number of instances by class is not perfectly balanced, containing mostly large instances. In this study we focus mainly on large instances, since as an starting point, large instances are the more critical to avoid collisions. The majority of models trained with this new labeled dataset were competitive to perform the cyclist's orientation detection, as can be appreciated in Fig. 6, except for the class CyclistSE, which fails mainly because the detectors frequently confuse it with CyclistSW. In order to correct this issue, tracking techniques are employed to filter abrupt changes in the detection and ensure time consistency.

For cyclist's orientation detection, the COCO metrics are presented in Table 4, in general Faster R-CNN meta-architecture with ResNet50 feature extractor obtained the best results for this multi-class detection. Surprisingly,

Table 5 Average precision with threshold AP@.50 by IoU using the CIMAT-CyclistV2 dataset for cyclist orientation detection

CyclistE	CyclistN	CyclistNE	CyclistNW	Architecture	Feature extractor
0.753154	0.716062	0.453501	0.506868	SSD	MobilenetV2
0.789926	0.842027	0.622228	0.649831	FasterRCNN	ResNet101
0.69774	0.809508	0.581257	0.604005	FasterRCNN	InceptionV2
0.781269	0.72074	0.542863	0.570648	SSD	InceptionV2
0.794934	0.763203	0.544258	0.625844	RFCN	ResNet101
0.763053	**0.852115**	0.577617	0.577302	**FasterRCNN**	**InceptionResNetV2**
0.808598	0.828415	**0.657107**	**0.686987**	**FasterRCNN**	**ResNet50**
CyclistS	CyclistSE	CyclistW	CyclistSW	Architecture	Feature extractor
0.620308	0.358924	0.711381	0.599037	SSD	Mobilenetv2
0.797039	0.443562	0.719340	0.690445	FasterRCNN	ResNet101
0.773338	0.484934	0.672358	**0.695525**	**FasterRCNN**	**InceptionV2**
0.593385	0.275655	0.69774	0.557210	**SSD**	**InceptionV2**
0.761743	0.470963	0.678240	0.668205	RFCN	ResNet101
0.838598	0.491344	0.567436	0.658789	FasterRCNN	InceptionResNetV2
0.791745	**0.499857**	**0.746269**	0.674663	**FasterRCNN**	**ResNet50**

The three most efficient meta-architectures were Faster R-CNN with ResNet50, Faster R-CNN with Inception-ResNetV2, and Faster R-CNN with ResNet101, but also SSD with InceptionV2 managed to obtain relatively good results in most classes, which positions it as a good alternative in orientation detection
Bold characters aim to highlight the best result for every criterion

ResNet with 50-layers outperformed precision-wise ResNet with 101-layers. This suggests that deeper networks require more instances to work better. Besides, all models with Faster R-CNN meta-architecture managed good result for this task.

In summary, for cyclist's orientation detection, the evaluation suggests that Faster R-CNN meta-architecture with InceptionV2 feature extractor allows for a good trade-off between precision and time response, offering a good performance considering AP, AR, localization and classification loss, as stated in Table 4. Moreover, in terms of response time for the region-based methods considered in this study, FasterRCNN-InceptionV2 was the fastest. On the other hand, if the main objective is to obtain a fast model for real-time applications, and high speed is required, the best trade-off is obtained with SSD meta-architecture using MobilenetV2 feature extractor. This is a good strategy to detect the orientation of the cyclist, even if it is not the most precise, since it allows to obtain the notion of the cyclist's movement at high rate, and is suitable to be implemented embedded on a low-cost vehicle with limited computation. Furthermore, this can be improved when combined with tracking techniques, as will be presented in the next section. Additionally, this model achieves considerably faster detections compared to other strategies reported in the literature for orientation detection [18,19].

5 Monitoring the cyclist safety

SSD-MobilenetV2, trained with the CIMAT-CyclistV1 dataset, was found to perform well for large size instances, which are the more critical for accident prevention, besides this model is considered the most suitable for real-time detection, given its faster response, which makes it a good alternative to apply tracking techniques. Once the cyclist is detected, a bounding box and confidence score is provided at high rate by the model SSD-MobilenetV2. To be considered as a positive detection, a confidence score higher than 0.5 is established. Likewise, a rough estimation of their orientation is also obtained by an SSD-MobilenetV2 model, trained with CIMAT-CyclistV2 for multi-class orientation detection. Thereafter, a Kalman filter (KF) tracking algorithm is applied to each instance to incorporate temporal information, increasing the smoothness and robustness on the detection against false positive (FP) and false negative (FN) detections, as well as occlusions. There, the association of the detections and tracker instances is carried out by means of the Kuhn–Munkres (K–M) algorithm that makes use of the intersection over union (IoU) metric. Finally, an evaluation of the level of risk is performed to each cyclist by considering their estimated position on one of the five different risk zones defined in the image, along with intention of motion based on the heading angle. The overall strategy for monitoring the cyclists safety is depicted in Fig. 7.

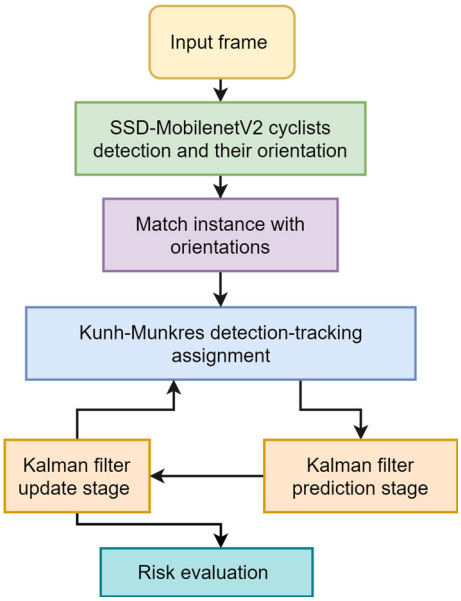

Fig. 7 Flowchart for detection and tracking of the cyclist in each frame

5.1 Cyclist tracking

For object tracking in the image, the Kalman filter is selected, which integrates information between frames to reduce the detection errors (FP and FN) and to address problems such as cyclist's occlusion [57]. The cyclists are tracked in the image plane through the bounding boxes coordinates, as provided by the detector, according to current and previous detections. Then, the aim is for the bounding box that covers the cyclist to be consistent over time, resulting in a smoother and more robust location estimation of the object of interest.

Let us consider a linear stochastic system in discrete time, with normal probability distribution of the form

$$\begin{aligned} \mathbf{x_k} &= \mathbf{F}\mathbf{x_{k-1}} + \omega_{k-1} \quad | \quad \omega \sim \mathcal{N}(\mathbf{0},\ \mathbf{Q}) \\ \mathbf{z_k} &= \mathbf{H}\mathbf{x_k} + \nu_k \quad | \quad \nu \sim \mathcal{N}(\mathbf{0},\ \mathbf{R}) \end{aligned} \tag{4}$$

where $\mathbf{F}$ stands for the transition matrix, while $\mathbf{H}$ is the measurement matrix. Also, ω and ν are the process and measurement noises, respectively, following a zero mean Gaussian distribution with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Subindex k denotes the time index, and would be omitted in the following for simplicity. The state vector $\mathbf{x}$, in time contains the coordinates of the bounding box in the image, defined by the center of the box (x, y), the width and height (w, h) and the heading angle (θ), as well as their first-order time derivatives, that is to say

$$\mathbf{x} = [x, \dot{x}, y, \dot{y}, w, \dot{w}, h, \dot{h}, \theta, \dot{\theta}] \tag{5}$$

and the measurement vector $\mathbf{z}$ is given by the bounding box parameters provided by any single-class detector in Table 2, and the heading angle provided by the multi-class detector

$$\mathbf{z} = [x_m, y_m, w_m, h_m, \theta_m] \tag{6}$$

here subindex m denotes measured data. Assuming that the variables (x, y, w, h, θ) are independent, and considering the second-order kinematic equation in discrete time, the relationship between a variable $\xi \in \{x, y, w, h, \theta\}$ and its time derivative $\dot{\xi}$ is approximated by

$$\begin{bmatrix} \xi_k \\ \dot{\xi}_k \end{bmatrix} \approx \begin{bmatrix} \xi_{k-1} + \dot{\xi}_{k-1}\Delta t + \frac{1}{2}\ddot{\xi}\left(\Delta t^2\right) \\ \dot{\xi}_{k-1} + \ddot{\xi}_{k-1}\Delta t \end{bmatrix} \tag{7}$$

where Δt is the time increment and $\ddot{\xi}_{k-1}$ is the second-order time derivative at the previous time step $k-1$. Then, a constant velocity model is used, where acceleration is considered as uncertainty; hence, the transition matrix $\mathbf{F}$ takes the form

$$\mathbf{F} = \begin{bmatrix} \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} & \cdots & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & \cdots & \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \end{bmatrix} \tag{8}$$

Meanwhile, the process noise $\boldsymbol{\omega}$ represents all the model uncertainties. Considering the second-order kinematic equation in Eq. (7), and the acceleration as an uncertainty, we can propose the process covariance matrix $\mathbf{Q}$ of the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{\boldsymbol{\xi}} & \cdots & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & \cdots & \mathbf{Q}_{\boldsymbol{\xi}} \end{bmatrix} \tag{9}$$

where, for a random variable $\xi \in \{x, y, w, h, \theta\}$, $\mathbf{Q}_{\boldsymbol{\xi}}$ is given by [2,3,42]

$$\mathbf{Q}_{\boldsymbol{\xi}} = \begin{bmatrix} \left(\frac{\Delta t^2}{2}\right)^2 & \left(\frac{\Delta t^2}{2}\right)\Delta t \\ \Delta t\left(\frac{\Delta t^2}{2}\right) & \Delta t^2 \end{bmatrix} \tag{10}$$

To determine the values of the measurement noise covariance matrix $\mathbf{R}$, the following procedure was followed, taking advantage of the subset for testing in the dataset presented in Sect. 3, containing ground truth labels for the bounding boxes. First, the selected model is run on the CIMAT-CyclistV1 test subset to obtain the bounding boxes predicted by the model. Then, for each instance i of the N cyclists in the test dataset, the ground truth bounding box $\mathbf{B}^i_{gt} = (x^i_{gt}, y^i_{gt}, w^i_{gt}, h^i_{gt})$ and measured bounding boxes $\mathbf{B}^i_m = (x^i_m, y^i_m, w^i_m, h^i_m)$ are used to obtain the errors $\mathbf{B}^i_e =$

$(e_x^i, e_y^i, e_w^i, e_h^i)$ for each of the bounding box parameters, such as

$$\mathbf{B}_e^i = \mathbf{B}_{gt}^i - \mathbf{B}_m^i \tag{11}$$

hence, for any elements of the error tuple $e_a, e_b \in \mathbf{B}_e$, the covariance of the error is given by

$$\sigma_{a,b} = \mathrm{Cov}(e_a, e_b) = \frac{1}{N}\sum_{i=1}^{N}(e_a^i - \bar{e}_a)(e_{e_b}^i - \bar{e}_b) \tag{12}$$

where $\bar{e}_a$ and $\bar{e}_b$ are the average of the errors over all the instances in the dataset corresponding to any parameters $a, b \in \{x, y, w, h\}$. N is the total number of cyclists in the test dataset. It is important to point out that the images in the dataset have different dimensions; hence, a normalization step must be performed before computing the covariances. Finally, the measurement noise covariance matrix $\mathbf{R} = \mathbf{R}^T$ takes a symmetric form.

5.2 Kuhn–Munkres algorithm for multiple target tracking

Multiple target tracking is acknowledged in this work by means of the Kuhn–Munkres (K–M) algorithm [27], a combinatorial optimization algorithm that solves the assignment problem in polynomial time. In the context of object detection, it has been used as part of the process of matching identified objects in time k to unidentified objects in time $k+1$ [41]. Hence, this algorithm can associate an object from one frame to another, based on a score, in this case the IoU presented in Eq. (2). Then, a data assignment or association step is performed to match each detection D to its respective instance of the tracker T.

Once the detection model generates a bounding box for each detected cyclist, the Kuhn–Munkres assignation cost matrix calculates the IoU distance between each detection and all the expected bounding boxes of the existing targets [4]. At each time step, the SSD-MobilenetV2 model returns a set of detections $\mathbb{D}_k = \{D_i | i = 0, 1, \ldots, m_k\}$, where D_i is a single instance detection and m_k is the total number of instances found in the image. In the same way, a set of trackers is generated for each cyclist in the scene $\mathbb{T}_k = \{T_j | j = 0, 1, \ldots, n_k\}$, where T_j is a single tracker and n_k is the number of existing trackers. The K–M algorithm is used to find the assignment which maximizes the IoU metric of trackers T and detections D. A cost square matrix $\mathbf{C}$ is defined with each element C_{ij} as the IoU resultant between the corresponding detection and tracker, if the number of detections and trackers is different, the remaining elements of the matrix are filled with zeros.

The matrix for this association process is shown in Fig. 8 as in [12,16]. This matrix allows us to identify which element in the detections $\mathbb{D}_k$ coincides better with which element in the trackers $\mathbb{T}_k$ (matched detections), which detection does not match with any existing tracker (unmatched detection), and which tracker does not match with any detection (unmatched tracker).

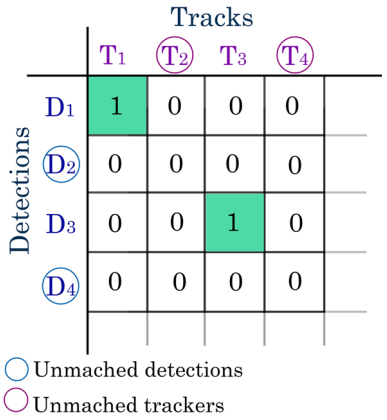

Fig. 8 Cost matrix for assigning a detection to track of Kuhn–Munkres algorithm (color figure online)

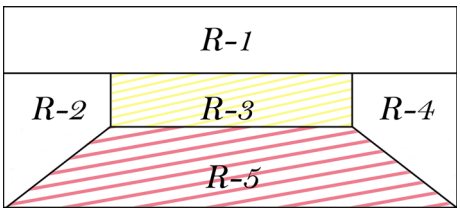

Fig. 9 Image regions to issue the cyclist vulnerability alert. The red zone R-5 represents the highest risk due to proximity. The yellow region R-3 is the warning zone. Lateral regions R-2 and R-4 may be risky or safe, depending on the cyclist orientation

At each time step, every detection in $\mathbb{D}_k$ is assigned to a tracker in $\mathbb{T}_k$, or in the case of an unmatched detection, a new tracker is added to $\mathbb{T}$ in the next iteration. Unmatched detections happen when the cyclist does not yet have a tracker assigned, mainly because it is the first time the cyclist appears on the scene. This allows us to attack the problem of FP detections, by determining a minimum number μ_1 of cyclist's detections before initializing a KF tracker. An unmatched tracker can also occur, in this case a target is not detected , mainly due to occlusions, FN detections or because it left the scene. To address this, a maximum number of time steps μ_2 is established to wait for the cyclist to reappear before a tracker is removed. A similar criterion is used independently for the head angle measurement, disregarding abrupt changes in orientation.

5.3 Risk level evaluation

Once a cyclist has been detected and tracked over time, it is also important to monitor its safety, that is, to determine

Fig. 10 Safety evaluation of multiple cyclists. The purple box is the measurement and the cyan box is the KF prediction. The cyclist orientation using SSD-MobilenetV2 is displayed with a purple arrow and the KF estimated angle with a cyan arrow. On top of each box, the green label indicates that the cyclist is not at risk, the yellow label represents that the cyclist is in moderate risk, and the red one signals imminent danger. **a**, **b** The case when cyclists are out of the way of the vehicle, but their orientation suggests their intentions of crossing to the danger region; hence, a *caution* yellow label is assigned. Finally, the blue circle stands for the Kalman filter's uncertainty **P**, as seen in (**e**) and (**f**), when the farther cyclists are occluded, but the KF continues to track them (color figure online)

when a cyclist is at risk of suffering an accident. Then, an alert can be emitted to the driver or to the ADAS system to slow down or stop the vehicle.

In this case, since only the use of a monocular camera is considered, without any depth information, similar to [35] the image is divided into 5 regions $\{R-1, \ldots, R-5\}$ considering the perspective, as shown in Fig. 9. The risk level is assigned to each cyclist based on the image location of the midpoint on the base of the corresponding bounding box, it is $(x_i + \frac{w_i}{2}, y_i + \frac{h_i}{2})$. Also, three levels of risk are taken into account *safe*, *caution* and *danger*.

Region $R-1$ indicates that the cyclist is far away from the camera and do not represent any risk; hence, a *safe* label is assigned. In the case of region $R-2$ and $R-4$, it corresponds to cyclists at the sides, which are considered out of the way of the vehicle, and a *safe* label may be assigned. However, special attention must be paid in this case due to the proximity to the camera, even if they do not represent an imminent risk, it could change suddenly if they intend to cross into the vehicle's way. Here the heading angle estimation is integrated in order to determine the cyclists intention of movement, and predict whether or not they will cross to the *danger* zone. For the $R-3$ region the VRUs are considered to be in the way of the vehicle, but at a considerable distance, indicating that the vehicle can proceed with *caution*. Finally, the region of greatest risk for the cyclist is region $R-5$, which indicates that it is too close to the camera, in this case the assigned label is *danger*.

6 Experimental results

In order to analyze the performance of the proposed strategy for monitoring the safety of VRUs, several experimental case studies were carried out in real urban environments, showing good results. In the following, some of these results are presented along with the implementation details.

Anytime μ_1 new unmatched detections are identified by the K–M algorithm, a new KF tracker is initialized with the first measured states and zero velocities. Moreover, the initial KF covariance matrix is set as $\mathbf{P}_0 = \mathbf{I}$. In addition, the covariance matrix of the process noise $\mathbf{Q}$ is selected as in Eqs. 9, 10, for a sampling period $\Delta t = 0.9s$. On the other hand, the uncertainty provided by the SSD-MobilenetV2 detector in the update stage, as described in Sect. 2.2, was determined by finding the covariance of the error between the predicted bounding boxes against the ground truth using the CIMAT-CyclistV1 test subset, containing $N = 709$ cyclist instances; then, the measurement noise covariance matrix $\mathbf{R}$ was found to be

$$\mathbf{R} = \begin{bmatrix} 137.65 & 8.90 & 24.43 & 8.22 & 0 \\ 8.90 & 83.84 & 7.89 & 3.74 & 0 \\ 24.43 & 7.89 & 174.43 & 10.71 & 0 \\ 8.22 & 3.74 & 10.71 & 64.01 & 0 \\ 0 & 0 & 0 & 0 & (\frac{\pi}{4})^2 \end{bmatrix} \tag{13}$$

where the heading angle noise is considered to be independent of the rest of the measurements, and its standard deviation is selected from the detector resolution of $\pi/4$.

As part of the K–M association algorithm, a maximum of $\mu_2 = 10$ consecutive unmatched detections are set before deleting a tracker, while a minimum of $\mu_1 = 4$ consecutive detections are considered before creating the tracker. Furthermore, the threshold for assigning a correspondence between a tracker and a detector was set to $IoU \geq 0.3$. Then, the KF allows to correct problems identified with the detection model, as it was identified that SSD-MobilenetV2 was not the most precise; nevertheless, it was the fastest detector, in this case running in average at 60 FPS, which allows to update the KF equations in real time and is suitable for future implementation embedded on a vehicle.

Various cases of the detection and tracking of cyclist under different scenarios are depicted in Fig. 10. These images were recorded in video at a resolution of 1920×1080 pixels at 30 FPS. Moreover, the performance of the proposed strategy can also be observed in video at https://youtu.be/JY78ZTcCuYM. There, in order to display the results, the bounding boxes are depicted in purple color for the detections from the SSD-MobilenetV2 , while the detected and estimated orientations are shown with arrows, and the KF predicted states are depicted as a cyan color box. From there, we can observe the effect of the KF over the detector, resulting in a smoother and more robust estimation with respect to occlusions and FP. Also, on top of each instance, a color label is displayed according to the level of risk, red for *danger*, yellow for *caution* and green for *safe*, showing satisfactory behavior. Finally, the KF uncertainty is also presented in the images as a blue ellipse, which can be observed to grow up when an instance is occluded and the tracker can only run the prediction stage without measurement updates.

From Fig. 10, we can analyze some interesting aspects of the proposed algorithms, showing satisfactory performance. Figure 10a presents a case where different levels of risk are detected, the closer cyclist is located in the highest risk region $R-5$ and a *danger* red tag is assigned, while the next instance is in the conditionally safe zone $R - 2$, but showing intentions to cross to the *danger* zone, hence a *caution* yellow tag is displayed, and the furthest away instance is considered to be *safe*. Multiple target tracking was successfully acknowledged using the KF plus K–M approach, as can be observed in Fig. 10e, f, where the KF is able to keep tracking occluded cyclists even when the detector fails, as seen in Fig. 10e, f nonetheless, since only the prediction step is used in those cases, the uncertainty (blue circle) is incremented over time, until the cyclist is detected again, or until the cyclist is considered to be gone and the tracker instance is deleted. Figure 10b, d presents cases when the cyclist is out of the way of the vehicle in the conditionally safe region $R - 2$, in the former, the estimated orientation of the cyclist suggest its intentions to cross to the danger zone; hence, a *caution* yellow tag is assigned, while in the latter, the cyclists intention is to pass beside the camera, and it is considered to be *safe* (green tag).

7 Conclusion and future work

In this work, we propose a full strategy to improve the safety on the road of a particularly vulnerable kind of VRU in the context of ITS. The strategy consists of a CNN-based detector coupled with multi target tracking with Kalman filters, and the Kuhn–Munkres assignment algorithm. It is important to note that the proposed method can be easily extended to other kinds of VRUs such as pedestrians and motorcycles.

Then, we propose a multi-class object detection technique based on the state-of-the-art CNN meta-architectures and feature extractors, where in addition to only detecting the object and its position, the detector also provides its orientation. Finally, this information is used to determine whether a VRU is intending to cross to a risk region, which may be crucial to prevent an accident.

In order to accomplish the multi-class orientation detection, we provide a new cyclist image dataset "CIMAT-Cyclist", which contains 20,229 cyclist instances over 11,103 images, labeled according to their orientation. Besides, we

extensively compare the state-of-the-art meta-architectures SSD, Faster R-CNN and R-FCN, combined with MobilenetV2, InceptionV2, ResNet50, ResNet101 and InceptionResNetV2 feature extractors for cyclist's and their orientation detection.

Experimental results in real scenarios suggested that the proposed strategy for monitoring the safety of the cyclists, is a simple but effective alternative to protect them on the road, timely providing useful information that may be used to alert the driver, or to automatically perform safety maneuvers in autonomous vehicles, such as slowing down, passing the cyclist or stopping the car. In this sense, it was shown that the use of a fast detector such as SSD-MobilenetV2 in combination with the multi target tracking algorithm provides a suitable solution for real time tracking of VRUs, since the tracking algorithm helps considerably to reduce the effects produced by errors and failures in the detector, while allowing for a faster response.

Future efforts will be dedicated to extend and balance the provided dataset, increasing the number of instances by class and size, in order to further improve the detectors performance, specially for similar classes such as CyclistSE vs CyclistSW. Also, increasing the number of smaller instances will help to better detect further away objects. In addition, it would be interesting to explore other modern DL techniques to improve the safety monitoring of VRUs, for example, using multi-view enhancement hashing [55] to improve the detection, or use novel techniques for scene understanding, such as detecting the road's topology in a similar way as in [56].

Also, for the tracking stage, it is intended to use an EKF with a more precise model considering the perspective transformation and the kinematics of the bicycles on the ground. Moreover, it would be useful to incorporate the orientation information in a tightly coupled scheme.

Finally, it is envisioned to further determine the cyclist movement-intention by detecting standard cycling hand signals.

Acknowledgements We thank the Mexican National Council of Science and Technology CONACyT for the grants given and the FORDECyT project 296737 "Consorcio en Inteligencia Artificial" and also "Sportbike Jerez MTB" for allowing us to take pictures at their events.

References

1. Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., et al.: Object detection through modified YOLO neural network. Sci. Program. **2020** (2020)
2. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. Wiley, Hoboken (2004)
3. Basso, G.F., De Amorim, T.G.S., Brito, A.V., Nascimento, T.P.: Kalman filter with dynamical setting of optimal process noise covariance. IEEE Accessed **5**, 8385–8393 (2017)
4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468. IEEE, Phoenix Convention Centre, Phoenix, Ariz (2016)
5. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3D object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems, pp. 424–432 (2015)
6. Chen, Y.Y., Jhong, S.Y., Li, G.Y., Chen, P.H.: Thermal-based pedestrian detection using faster R-CNN and region decomposition branch. In: 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 1–2. IEEE, Taipei, Taiwan (2019)
7. Cho, H., Rybski, P.E., Zhang, W.: Vision-based 3D bicycle tracking using deformable part model and interacting multiple model filter. In: 2011 IEEE International Conference on Robotics and Automation, pp. 4391–4398. IEEE, Shanghai (2011)
8. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems 29 (NIPS 2016), pp. 379–387. Barcelona, Spain (2016)
9. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012). https://doi.org/10.1109/TPAMI.2011.155
10. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2155–2162. Columbus, OH, USA (2014)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
12. Ferguson, M., Law, K.: A 2d-3d object detection system for updating building information models with mobile robots. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1357–1365. IEEE (2019)
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE, Providence, Rhode Island, USA (2012)
14. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448. Santiago, Chile (2015). https://doi.org/10.1109/ICCV.2015.169
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587. Columbus, OH, USA (2014). https://doi.org/10.1109/CVPR.2014.81
16. Guan, K.: Computer vision based vehicle detection and tracking using tensorflow object detection api and kalman-filtering (2018)
17. Guindel, C., Martín, D., Armingol, J.M.: Modeling traffic scenes for intelligent vehicles using CNN-based detection and orientation estimation. In: ROBOT 2017: Third Iberian Robotics Conference, pp. 487–498. Springer, Seville, Spain (2017)
18. Guindel, C., Martin, D., Armingol, J.M.: Fast joint object detection and viewpoint estimation for traffic scene understanding. IEEE Intell. Transp. Syst. Mag. **10**(4), 74–86 (2018)
19. Guindel, C., Martin, D., Armingol, J.M.: Traffic scene awareness for intelligent vehicles using ConvNets and stereo vision. Robot. Auton. Syst. **112**, 109–122 (2019)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. Las Vegas, NV, USA (2016)

21. Heo, D., Nam, J.Y., Ko, B.C.: Estimation of pedestrian pose orientation using soft target training based on teacher-student framework. Sensors **19**(5), 1147 (2019)
22. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3296–3297. IEEE, Honolulu, HI, USA (2017)
23. Hui, J.: Object detection: speed and accuracy comparison (faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOV3). Medium (2018). https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359. Accessed on 25 July 2019
24. Jonker, R., Volgenant, T.: Improving the Hungarian assignment algorithm. Oper. Res. Lett. **5**(4), 171–175 (1986)
25. Jung, H., Tan, J.K., Ishikawa, S., Morie, T.: Applying hog feature to the detection and tracking of a human on a bicycle. In: 2011 11th International Conference on Control, Automation and Systems, pp. 1740–1743. IEEE, Gyeonggi-do, Korea (South) (2011)
26. Kang, Y., Yin, H., Berger, C.: Test your self-driving algorithm: an overview of publicly available driving datasets and virtual testing environments. IEEE Trans. Intell. Veh. **4**(2), 171–185 (2019)
27. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955). https://doi.org/10.1002/nav.3800020109
28. Lan, W., Dang, J., Wang, Y., Wang, S.: Pedestrian detection based on YOLO network model. In: 2018 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1547–1551. IEEE, Changchun, China (2018)
29. Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., Gavrila, D.M.: A new benchmark for vision-based cyclist detection. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 1028–1033. IEEE, Gothenburg, Sweden (2016)
30. Li, X., Li, L., Flohr, F., Wang, J., Xiong, H., Bernhard, M., Pan, S., Gavrila, D.M., Li, K.: A unified framework for concurrent pedestrian and cyclist detection. IEEE Trans. Intell. Transp. Syst. **18**(2), 269–281 (2017). https://doi.org/10.1109/TITS.2016.2567418
31. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer, Zurich, Switzerland (2014)
32. Liu, C., Guo, Y., Li, S., Chang, F.: ACF based region proposal extraction for YOLOv3 network towards high-performance cyclist detection in high resolution images. Sensors **19**(12), 2671 (2019)
33. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: a survey. Int. J. Comput. Vis. **128**(2), 261–318 (2020)
34. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer, Amsterdam, The Netherlands (2016)
35. Maurya, S.K., Choudhary, A.: Deep learning based vulnerable road user detection and collision avoidance. In: 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 1–6. Madrid (2018)
36. Ohn-Bar, E., Trivedi, M.M.: Fast and robust object detection using visual subcategories. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 179–184. Columbus, OH, USA (2014). https://doi.org/10.1109/CVPRW.2014.32
37. Phon-Amnuaisuk, S., Murata, K.T., Pavarangkoon, P., Yamamoto, K., Mizuhara, T.: Exploring the applications of faster R-CNN and single-shot multi-box detection in a smart nursery domain. arXiv preprint arXiv:1808.08675 (2018)
38. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. Las Vegas, NV, USA (2016)
39. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. Honolulu, HI, USA (2017)
40. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
41. Sahbani, B., Adiprawita, W.: Kalman filter and iterative-Hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: 2016 6th International Conference on System Engineering and Technology (ICSET), pp. 109–115. IEEE, Bandung, Indonesia (2016)
42. Saho, K.: Kalman filter for moving object tracking: performance analysis and filter design. In: Kalman Filters-Theory for Advanced Applications, pp. 233–252 (2017)
43. Saleh, K., Hossny, M., Hossny, A., Nahavandi, S.: Cyclist detection in LIDAR scans using faster R-CNN and synthetic depth images. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE, Yokohama, Japan (2017)
44. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520. Salt Lake City, UT (2018)
45. Saranya, K.C., Thangavelu, A., Chidambaram, A., Arumugam, S., Govindraj, S.: Cyclist detection using tiny YOLO V2. In: Soft Computing for Problem Solving (SocProS), pp. 969–979. Springer (2020)
46. Sermanet, P., Kavukcuoglu, K., Chintala, S., Lecun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3626–3633. Portland, OR, USA (2013). https://doi.org/10.1109/CVPR.2013.465
47. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence. San Francisco, California, USA (2017)
48. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. Las Vegas, NV, USA (2016)
49. Tian, W., Lauer, M.: Fast and robust cyclist detection for monocular camera systems. In: 10th International joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP). Berlin, Germany (2015)
50. Tian, W., Lauer, M.: Detection and orientation estimation for cyclists by max pooled features. In: 12th International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), pp. 17–26. Porto, Portugal (2017)
51. Wang, K., Zhou, W.: Pedestrian and cyclist detection based on deep neural network fast R-CNN. Int. J. Adv. Robot. Syst. **16**(2), 1729881419829651 (2019)
52. World Health Organization: Global status report on road safety 2018. World Health Organization, Geneva (2018). https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/
53. World Health Organization: Road Safety 2018. https://extranet.who.int/roadsafety/death-on-the-roads/ (2020). Accessed: July 2020
54. Xu, J.: Deep learning for object detection: a comprehensive review. Towards Data Science (2017)
55. Yan, C., Gong, B., Wei, Y., Gao, Y.: Deep multi-view enhancement hashing for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1445–1451 (2020)

56. Yan, C., Shao, B., Zhao, H., Ning, R., Zhang, Y., Xu, F.: 3D room layout estimation from a single RGB image. IEEE Trans. Multimedia **22**(11), 3014–3024 (2020)
57. Yang, F., Chen, H., Li, J., Li, F., Wang, L., Yan, X.: Single shot multibox detector with Kalman filter for online pedestrian detection in video. IEEE Accessed **7**, 15478–15488 (2019)
58. Zhang, M., Fu, R., Guo, Y., Wang, L., Wang, P., Deng, H.: Cyclist detection and tracking based on multi-layer laser scanner. HCIS **10**, 1–18 (2020)
59. Zhang, S., Wang, X.: Human detection and object tracking based on histograms of oriented gradients. In: 2013 Ninth International Conference on Natural Computation (ICNC), pp. 1349–1353. Shenyang, China (2013). https://doi.org/10.1109/ICNC.2013.6818189
60. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)

M. García-Venegas was born in 1994 in Jerez de García Salinas, Zacatecas. She received her masters degree in Software Engineering from the Center for Research in Mathematics, CIMAT-Zacatecas, in 2020, with the thesis titled "Vulnerable road user safety by detection and tracking the cyclist using Deep Learning". She is currently working on software projects. Her areas of interest are object detection, computer vision, AI and software engineering.

D. A. Mercado-Ravell was born in Mexico City. He received his B.S. degree in Mechatronics Engineering from the Universidad Panamericana in Aguascalientes, Mexico, the M.Sc. degree in Electrical Engineering option Mechatronics from CINVESTAV-IPN, Mexico City, and the Ph.D. in Automation, Embedded Systems and Robotics from the University of Technology of Compiègne, France. Dr. Mercado-Ravell has held postdoctoral positions at the Mechanical and Aerospace Department at Rutgers, the State University of New Jersey, USA, and at CINVESTAV Mexico. He is currently full-time professor at CIMAT-Zacatecas, in Mexico, and member of the national research system (SNI) since 2018. His research topics include robotics, modeling and control, unmanned aerial/underwater vehicles, autonomous navigation, data fusion, computer vision and deep learning applications.

L. A. Pinedo-Sánchez received his masters degree in Software Engineering from the Center for Research in Mathematics, CIMAT-Zacatecas in 2020. He has been working on bearing vibration analysis for failure prevention using convolutional neural networks. He is currently a software developer. His research interests include deep learning and machine learning.

C. A. Carballo-Monsivais is a data scientist, who collaborates in the Data Engineering Group of the Center for Research in Mathematics CIMAT A.C. Zacatecas Unit. He has held consultancies in Mexico, Ecuador, Cuba, Colombia, Argentina, Chile and the United States. Being a Master Trainer by Plexus International in events sanctioned by AIAG (Automotive Industrial Action Group), he has more than 18 years of experience in Design for Six Sigma, Lean Six Sigma, Industrial Statistics, Big Data, Problem Solving and Technology Transfer, Machine Learning, Deep Learning, Data Science, Project Management with Kanban and Scrum.