IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

Pedestrian Street-Cross Action Recognition in Monocular Far Infrared Sequences

RALUCA DIDONA BREHAR[1], MIRCEA PAUL MURESAN[1], TIBERIU MARIŢA[1], CRISTIAN-COSMIN VANCEA[1], MIHAI NEGRU[1], and SERGIU NEDEVSCHI[1], (Member, IEEE)

[1]Technical University of Cluj-Napoca
Department of Computer Science

Corresponding author: Sergiu Nedevschi (e-mail: Sergiu.Nedevschi@cs.utcluj.ro)

This work was supported by: GNaC 2018 ARUT grant "Object detection in FIR thermal monocular images for night vision applications", research Contract no. 3091/05.02.2019; "CARSafe" a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0917, contract 21 PCCDI/2018, within PNCDI III; and by the European Fund of Regional Development through the Competitiveness Operational Programme 2014-2020, CLOUDUT Project, contract no. 235/2020.

ABSTRACT The early recognition and understanding of the actions performed by pedestrians in traffic scenes leads to an anticipation of pedestrian intentions in advance and helps in the process of collision warning and avoidance in the context of autonomous vehicles. An environment with low visibility conditions such as night-time, fog, heavy rain or smoke increases the number of difficult situations in traffic. A complete and original model for assessing if a pedestrian is engaged in a street cross action using only infrared monocular scene perception is proposed in this paper. The assessment of a street cross action is done by the time series analysis of features like: pedestrian motion, position of pedestrians with respect to the drivable area and their distance with respect to the ego-vehicle. The extraction of these features emerges from the combination of a deep learning based pedestrian detector with an original tracking algorithm, a semantic segmentation of the road surface and a time series long-short term memory network based action recognition. In order to validate the proposed method we introduce a new dataset named CROSSIR. It is formed of pedestrian annotations, action annotations and semantic labels for the road. The CROSSIR dataset is suitable for several common computer vision algorithms: (1) pedestrian detection and tracking algorithms because each pedestrian has a unique identifier over the frames in which it appears; (2) pedestrian action recognition; (3) semantic segmentation of the road pixels in the infrared image.

INDEX TERMS Image Processing, Neural Network, Pattern Recognition, Night Vision Applications, FLIR Camera, Pedestrian Detection, Pedestrian Tracking, Semantic Segmentation, Time Series Analysis

I. INTRODUCTION

NUMEROUS approaches that are able to achieve state of the art results for pedestrian action, intention and behavior recognition are present in the active field of computer vision for autonomous vehicles [1].

Existing pedestrian action and intention recognition solutions address mostly information extracted from color or gray scale images [1], [2], [3] that come from monocular or stereo-vision camera setups suitable in particular for daylight driving scenarios. Little is explored for the situation of night traffic scenes. For these particular situations, cameras that capture the heat emitted by objects can be used. Far infrared sensors are suitable for night driving situations. The development of algorithms coping with the information provided by far infrared cameras provides a promising field of research and can lead to robust and accurate solutions for pedestrian detection, tracking and pedestrian action recognition.

The method proposed in this paper addresses the problem of street cross action recognition in the framework of a monocular far infrared setup in which images have been captured during winter and spring, both in day and night driving situations.

Figure 1, presents a set of states that a pedestrian transits while performing a street cross action. It represents a classical situation in which the pedestrian comes towards the street and keeps crossing without stopping.

As it can be noticed in Figure 1-a the pedestrian is crossing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

(a) Continuous cross action. The temporal order of the frames is from left to right.

(b) Not cross: walk towards the road then stop. The temporal order of the frames is from left to right

FIGURE 1: Sequence of states for a cross vs. not cross action

(from right to left) at a marked place but the zebra marking is slightly visible in the infrared image. Hence approaches based on the position of a subject with respect to a marked crossing are not applicable in the case of the infrared field.

Another characteristic situation in assessing the awareness of an autonomous driving system with respect to the cross action performed by a pedestrian involves a continuous motion towards the drivable area, followed by a stop in motion, meaning the pedestrian has a low probability of crossing the street. This situation is depicted in Figure 1-b.

The two situations exemplified in Figure1 are frequently encountered in every day driving scenarios. A high awareness of the autonomous driving system regarding the actions performed by pedestrians, especially the cross versus not cross situations, would improve the system anticipation level and furthermore could reduce the chance of injury.

The proposed solution is based on the interaction and interconnection of several coupled modules which all define a multi-cue environment representation model. The main components are:

1) Data acquisition;
2) Road surface estimation;
3) Pedestrian detection and tracking;
4) Pedestrian distance estimation;
5) Pedestrian speed computation;
6) Pedestrian action recognition;

The authentic outcomes and contributions of the paper reside in:

- The proposal, design and development of an original tracking algorithm applied on top of a deep learning based pedestrian detector fine tuned to work with far infrared images.
- The recognition of pedestrian street-cross or not cross actions, in the difficult situations of night and / or low visibility driving. A time series Long Short Term Memory based model is trained using features like:
  - -- Distance of the pedestrian with respect to the ego-vehicle (in meters)
  - -- Pedestrian motion features like the horizontal and vertical optical flow components and the horizontal speed component (transversal to the road) of the tracked pedestrian.
  - -- Position of pedestrians with respect to the road: pedestrian on road or pedestrian off road.
- In order to validate the results of the proposed solution a dataset of cross / not cross annotated image sequences captured with a FLIR infrared camera is introduced. The experimental results report a recognition accuracy of 93%.

II. RELATED WORK

The proposed solution is a complete model that comprises pedestrian detection in infrared sequences, a robust tracking method and a cross action recognition module. A survey of existing state of the art methods is presented in the following subsections.

A. PEDESTRIAN DETECTION IN INFRARED IMAGES

Even though most of the pedestrian detection approaches are based on monocular or stereo cameras, the use of FLIR cameras has attracted attention of the research community and manufacturers of ADAS or autonomous systems due to the ability of thermal cameras to provide reliable detections for vulnerable road users in bad weather conditions such as snow, fog, rain or bad illumination situations.

Feature based classification approaches and deep learning models are highly explored in what regards the topic of detecting pedestrians in infrared images. Traditional classification techniques that extract visual image features which are fed to machine learning algorithms are highly explored in color images [4], [5], [6] and their typologies have been restructured and adopted for infrared images or for fused infrared and color images [7]. Histogram of Oriented Gradients, Local Binary Patterns, Edgelets or feature pyramids are combined with AdaBoost, Support Vector Machine and other types of algorithms in order to obtain reliable classification

solutions [8], [9], [10], [11], [12].

The high recognition rates achieved using deep learning based object recognition models [13] in color images constitute the basis for object detection and recognition in the infrared domain. An illumination aware Faster R-CNN deep learning based convolutional neural network architecture is employed by [14] for pedestrian detection in both infrared and color images. A brightness aware deep learning based mechanism is proposed by [15] and it is used to detect pedestrians under day or night conditions respectively. An automatic region proposal network is introduced by [16] to generate bounding boxes with confidence scores for far-infrared (FIR) pedestrian detection. A Faster R-CNN network is trained on infrared images augmented with their saliency maps that serve as an attention mechanism for the pedestrian detector [17]. The saliency maps are generated using static and deep methods and show an improvement in detection especially during daytime. A multi-class object detection solution based on YOLO [18] architecture is presented by [19] with a focus on pedestrian and car detection in monocular infrared images. To validate the model, the authors of [16] uses the LSI, CVC09, CVC14 and SCUT FIR pedestrian detection datasets in their experiments. They obtain a log average miss rate of 49.4 for CVC09, 38.06% for LSI and 17.54% for SCUT FIR.

B. PEDESTRIAN TRACKING

Multi object tracking (MOT) can be applied in very many different settings and scenarios, and for some advanced technical systems, like autonomous cars, multiple object tracking is a necessary enabling technology. For an autonomous vehicle to drive safely in an urban environment, it is important to track pedestrians or cyclists while using this information carefully to plan its trajectory for collision avoidance. The challenges that appear in the MOT problem can be split into two categories, sensor and data association related issues. The far infrared sensor related challenges may refer to:

- Unknown number of objects with unknown number of states that are present in the sensor field of view (FOV)
- Objects leave and enter the FOV of the sensor
- The detector of objects is imperfect and is susceptible to two kind of errors i.e. missed detections (due to environment conditions, object properties, occlusions) and false detections or clutter (a detection that is not caused by an object). Both error types can lead to fatal outcomes in the worst-case scenario if they are not handled.

In addition to the sensor challenges stated above there is yet another challenge in target tracking which is the called the data association problem. The gist of this problem is that we do not have any information about the origin of the detection or what caused them. Therefore, the challenges for treating the data association problem can be split into two categories:

- The origin uncertainty âĂŞ we do not have any information about the new measurements and how they relate to the previous sensor data
- The motion uncertainty âĂŞ objects can have multiple motion models

Poor handling of the data association problem leads to bad tracking results. Most multi object tracking methods use a tracking-by detection framework, which means they rely on an object detector to provide the object candidates. Multiple papers in the literature propose tracking solutions that address the previous mentioned problems using different types of sensors: like single cameras, stereo cameras LIDARS, RADAR, FIR (far infra-red) or a combination of them. Several studies describe tracking systems in video sequences taken by color or monochrome cameras mounted on a vehicle [20], [21]. Some approaches use a handcrafted cost function [22], [23], allowing a better control over the selected features and the data association process, while other methods propose deep learning (or data-based) association and tracking methods [24], [25] which let a neural network decide the best feature combination for solving the correspondence problem. The main issue with deep learning and data-based methods in general is that the tracker may get latched onto an object, that may be a false detection, but looks similar to something from the training data-set, and never recover. Furthermore, if motion information is not incorporated in the neural network model, in case an object is occluded by a similar object the tracker may get latched onto the wrong object.

When fusing multiple sources of information for performing more robust object tracking, very many of the current approaches center their solution on a single input (like the camera) or do not exploit the information coming from all the input sources [26], [27]. This means that in case of camera failure their solution would not function properly. The authors in [28] and [29] address this issue by ensuring that each sensor is able to perform its role reliably and independently. The overall system performance is improved when all sensors are functioning, however in case one sensor is not working, the whole system does not crash. The solution in [28] uses deep learning to fuse the different modalities, while the method presented in [29] uses a combination between an Unscented Kalman Filter (UKF) and single layer perceptron to fuse the data. In another approach [30] the authors use deep neural networks to jointly detect and track 3D objects using a stereo camera system. In this approach a neural network is used to detect 2D bounding boxes in images, and improve the 3D bounding box detected from the point cloud using a regression strategy.

The use of Far Infrared(thermal) cameras has attracted many researchers due to their ability to operate in bad weather conditions and in low illumination or night conditions. Some approaches describe solutions using single thermal [31] cameras for tracking, while others use stereo vision [32], based on far infrared cameras, to reconstruct and track pedestrians. Other solutions combine probabilistic algorithms for tracking pedestrians using a thermal camera. For example, in [33] the authors use a Kalman filter and a mean shift algorithm to find the exact position of mov-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

ing pedestrians. In [34] the authors fuse multiple sensors including a LIDAR and FIR camera to obtain both ego motion and distance estimation. The solution presented in [35] illustrates a modular approach for tracking pedestrians by merging the predictions of the Kalman filter with past history analysis. This solution is able to help correct temporary miss-recognitions that occur when the detector fails as well as reduce false detections. The authors in [36] try to solve the data association and tracking problem in thermal images using deep network architectures. They propose a feature model comprising of thermal infrared specific features and correlation features for thermal infrared object representation. The features are coupled for a more robust data association and tracking using a multi task matching framework. The paper presented in [37] proposes a simple weighted function that combines similarities in position, size and appearance. The main issue with this work is that the appearance score is computed in a naive manner and no information related to the motion of the pedestrian is used in the final cost function, which would make the data association fail in case of similar overlapping pedestrians.

C. PEDESTRIAN ACTION RECOGNITION

The development of methods able to estimate pedestrian's action of crossing the street is an active field of research, especially nowadays when the autonomous vehicles are starting to be part of an every-day reality in urban traffic. A detailed survey of existing approaches in pedestrian action recognition and intention prediction is performed by [38]. As it is described, even if largely addressed in the scientific community, the pedestrian intention prediction subject is a challenging problem because pedestrians have an unpredictable behavior, they can move in any direction and suddenly change motion [38]. Existing approaches consider particular situations like crossing at an intersection, or at a marked crossing zebra, or more generically, in situations when the street is not marked at all.

Several categories of prediction models are very popular [39]: (1) pedestrian related approaches (2) context based approaches (3) path prediction approaches. These approaches are applied on color images and few methods have been proposed for infrared images.

Pedestrian related methods define a model in terms of features (motion, appearance) and these features are learned by a classifier. For example, numerous approaches consider features that define the body pose of pedestrians by means of skeletons or 2D joints. [40] use a CNN based model for skeleton fitting (pose estimation) and the most stable points of the skeleton which correspond to the legs and the shoulders are fitted to a Random Forest (RF) classifier that provides the probability of the cross / not cross action. [2] use a low dimensional feature vector that contains flow variations on the pedestrian legs and upper body. Stereo measurements, vehicle velocity and yaw-rate measurements are considered to compute the ego-motion compensated and normalized optical flow field that is further used to extract features given a bounding box detection and distance estimation z of a pedestrian. The action classification is done using a particle filter model. The 2D articulated pose extracted by a convolutional neural network model from monocular images is employed by [40] in order to recognize the intentions of both pedestrians and cyclists. A Random Forest (RF) classifier is applied on top of skeleton features and it provides a probability to perform the cross vs. not cross classification. They consider only pedestrian training samples with a minimum bounding box width of 60 pixels and no occlusion.

Context related approaches integrate pedestrian features with environment clues. The authors of [41] propose a descriptor based on the motion of the pedestrian relative to the road and based on the spatial layout of the scene considering information like pedestrian lights, zebra crossings and traffic islands, waiting areas as bus stops. A classification based on Support Vector Machine is applied to the feature vector for predicting the pedestrian intention in color intensity images. An algorithm that predicts the pedestrian's intention to cross the street in infrared images is presented in [42]. Dynamic fuzzy automata are employed in combination with spatio-temporal features like the distance between the curbs and the pedestrian, the velocity of pedestrians and head orientation. Furthermore, the authors consider four intention states for pedestrians: standing-sidewalk, walking-sidewalk, walking-crossing and running crossing. The predicted intentions are 'stop' or 'cross'.

Path prediction approaches are highly related to intention estimation besides current action recognition. For example, [43] propose an encoder-decoder Long Short-Term Memory (LSTM) network that extracts the state streams from both vehicle trajectory and pedestrian trajectory. A decoder network performs the state fusion and predicts the future trajectory. The pedestrian location and pose are inferred by means of a Balanced Gaussian Process Dynamical Model (B-GPDM) and naÃŕve-Bayes classifiers in the work of [44]. The classifiers use 3D joint positions in lateral direction and the displacements of the 3D joints in the same direction. Based on the lateral position of a pedestrian [45] a long-term intent prediction model is proposed. They train a stacked LSTM and formulate the intention as a time series prediction problem.

The quantification and labeling of the pedestrian crossing intention depends on the type of intention model proposed. Several types of annotations and labels for pedestrian crossing intentions have been explored. For example [41] define the pedestrian crossing intention in relation with the situation when the pedestrianâĂŹs principal aim is to cross the street. A human based annotator rates the cross intention in an interval from 0 to 1 with a step of 0.25, where 0 means the pedestrian does not cross the street, and 1 means the pedestrian crosses the street. The inner intervals of 0.25, 0.5 and 0.75 model possible uncertainties upon the pedestrian decision.

Two scenarios are considered by [2] when the pedestrian is walking towards the road side curb: will the pedestrian cross

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

or stop at the curb. For each trajectory where the pedestrian stops the moment of the last placement of the foot is labeled as the stopping moment. A time-to-stop value is set to zero for that moment. Similarly for cross scenarios a time-to-curb is defined in relation with the closest point to the curbstone (with closed legs).

The authors of [40] have enriched the JAAD dataset [46] with time-to-event (TTE) information for the cross actions. Two scenarios were considered by [40]: start-walking-to-cross when the pedestrian stands near the road and then he starts to cross and keep-walking-to-cross when the pedestrian is involved in a continuous cross action. For keep-walking-to-cross the time to event is zero at the first frame at which the trunk of the walking pedestrian is over the curbside. For start-walking-to-cross the time to event is zero at the frame at which the stopped pedestrian starts moving a leg forward. Positive TTE values correspond to frames before the event, negative values to frames after the event.

The work on cross action recognition is based on pedestrian detection algorithms, which rely on benchmark datasets for infrared images. The most popular datasets are:

- KMU Pedestrian Intention Prediction Database in Thermal Images [42] contains a collection of infrared sequences captured by a FIR camera of a moving car roof at nighttime. The dataset totals 3254 frames and 37 pedestrians, collected in 6 videos. Each video shows four behaviours: standing, walking on the sidewalk, walking, and running on the road.
- The KAIST MultiSpectral Pedestrian Dataset was introduced by [7]. It comprises pedestrian annotated instances for pairs of temporally and spatially aligned color and infrared images, corresponding to both day and night situations.
- SCUT FIR Pedestrian Dataset [47] consist of about 11 hours-long image sequences at a rate of 25 Hz by driving through diverse traffic scenarios at a speed less than 80 km/h. Bounding box annotations are provided for 7,659 unique pedestrians.
- FLIR-ADAS [48] provides multi-class annotations for far infrared images. The instance labels are for pedestrians (over 20k annotations), cars, bicycles and dogs.
- PTB-TIR: A Thermal Infrared Pedestrian Tracking Benchmark [49] that contains sequences of thermal images, which are annotated manually. The benchmark also contains the results and rankings of different tracking algorithms on the provided image datasets.

It can be noted that only the KMU [42] pedestrian dataset has annotations that support cross action recognition algorithms. The dataset is suitable for pedestrian detection based algorithms as it contains action and bounding box annotations. With this paper we enrich the field of benchmark datasets by introducing CROSSIR action recognition dataset that can be used for pedestrian, context and motion based approaches. It contains bounding box annotations, unique identifiers for pedestrians, semantic segmentation information for road pixels and motion type descriptors (walk, stand, run). The dataset is available for the scientific community [1].

As it can be noted from existing state of the art approaches, the pedestrian cross action recognition for color images is highly explored in the literature. However, approaches to infrared based pedestrian cross action recognition are not yet sufficiently addressed. The model proposed in this paper combines pedestrian feature with context and motion based approaches. It uses a deep learning based pedestrian detector and a novel texture based tracking approach that ensures stable detections across successive frames and provides motion information along with pedestrian features. All these features are combined with context information which is extracted by the combination of a semantic segmentation of the road and monocular distance estimation which enhance the pedestrian feature vector which is used by a Long Short Term Memory Network for recognizing the cross action in infrared scenes.

III. MATERIALS

In order to ensure variety for the experiments presented in this paper, and to cover day and night diving scenarios using a far infrared camera, a new dataset is introduced: CROSSIR. It contains annotated infrared sequences, with focus on cross / not cross actions of pedestrians, but also a set with road segmentation ground truth which is applicable for semantic segmentation.

A. ACQUISITION SYSTEM

The used image sensor consists of a FLIR PathFindIR camera, incorporating an uncooled 320x240 Vox microbolometer, with 8-14 μm spectral response. It is equipped with automatically heated 19mm lens providing a 36°(h) and 27°(v) field of view. The core is hermetically sealed and protected against dust and water spreads (IP67 rated) allowing the unit to perform in a wide range of weather conditions. The PAL analog video output running at 25 fps is turned into digital format with DVD EZMaker 7 converter from AVerMedia and the images are up-sampled to 640x480 resolution.

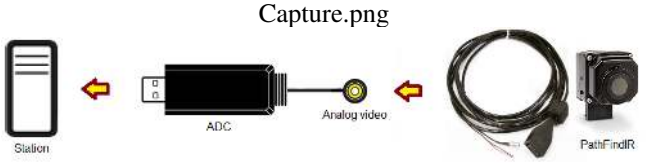


FIGURE 2: Aquisition System: the infrared camera, the analog to digital converter, the system that receives the captured frames.

B. CROSSIR DATASET

The proposed dataset contains sequences of infrared images grabbed in winter and spring time. Acquired frames have a resolution of 640x480 pixels. The annotation has been

[1] https://users.utcluj.ro/ raluca/crossir/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

realized using the Computer Vision Annotation Tool (CVAT) [50].

Pedestrian annotations are present for 86 sequences of various lengths captured during night or day in the city of Cluj-Napoca, Romania. An annotation contains:

- Pedestrian identifier (id) – that is unchanged for every frame in which the pedestrian appears, making the dataset appropriate for tracking algorithms.
- Pedestrian bounding box in the form of top left coordinates, width and height.
- A label for the performed action. It can be cross or not cross.
- A label for the direction of movement with respect to the road. The label can take the values: lateral, longitudinal or diagonal.
- A label for the type of motion: walk, stand, run.
- A label that marks if the pedestrian is occluded or not.

The cross scenarios captured by the proposed dataset are :

1) pedestrians walking or running towards the road and crossing continuously.
2) pedestrians standing close to the curb and starting to cross
3) pedestrians walking on the road, having a longitudinal direction of motion (their motion is parallel to the motion vector of the ego-vehicle)

These scenarios are depicted in Figure 3.

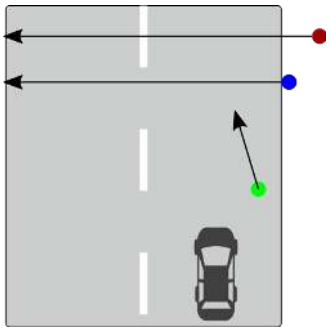


FIGURE 3: Scenarios for cross action: pedestrians are marked with a point: the red point figures a pedestrian that is walking or running towards the road and is crossing continuously without stopping, the blue point represents the scenario in which a pedestrian is standing close to the curb and starts to cross, while the green point represents a pedestrian walking on the road.

The not cross scenarios acquired in the proposed dataset are:

1) Pedestrian standing, walking or running parallel to the road. In this situation the pedestrians do not enter the drivable area and their direction of motion is parallel to the road.
2) Pedestrians walking or running towards the road and stopping.

These scenarios are depicted in Figure 4. The dataset contains

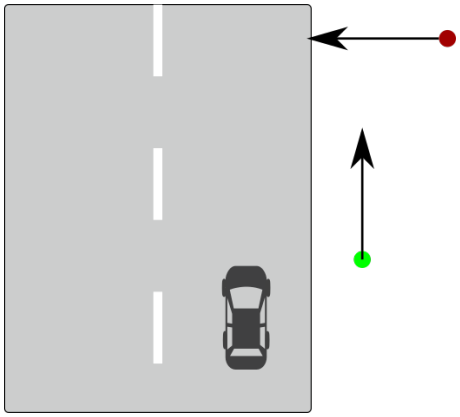


FIGURE 4: Scenarios for not cross action: the red point represents a pedestrian that walks towards the street and then stops, the green point represents a pedestrian walking on the pavement without entering the street area.

fully visible and also occluded pedestrians. At least one pedestrian is present in each sequence. The total number of annotated frames is 14678. The dataset contains a total number of 175 unique pedestrians. Road segmentation sequences contain 471 night frames, and 376 day frames. These are annotated as polygonal areas. The annotations are made for a random subset of frames from all the acquired videos in order to ensure the large diversity of the annotations.

IV. METHODS

The proposed processing pipeline is described in Figure 5 and its main modules are:

- Pedestrian detector – applies a CNN based detector to input images for each frame. Its outputs are a set of bounding boxes defined by position and size: $[x, y, width, height]$.
- Pedestrian tracking – performs tracking on top of detected bounding boxes. It improves quality of detections and provides an updated list of bounding boxes, speed components (vx, vy), optical flow magnitude and angle.
- Distance estimator – using the geometric constraints of the system setup (position of FLIR camera on top of the ego-vehicle) it computes the distance of any point in the image with respect to the camera and also with respect to the Ego vehicle coordinate system (centered in the road projection of the mid point of the front bumper). It is used for estimating the relative position and speed between the tracked pedestrians and the ego-vehicle.
- Action recognition module – uses a Long Short Term Memory Network that given a sequence of states and the measured features for a pedestrian in each state (frame) provides the probability the pedestrian is engaged in a cross or not cross action.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

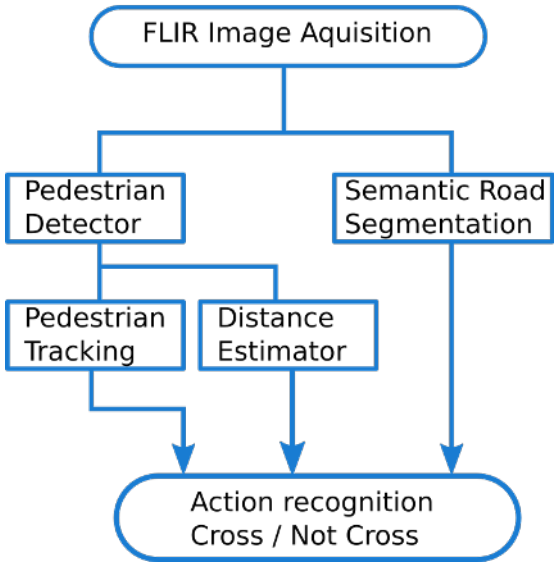


FIGURE 5: Main modules of the proposed processing pipeline: the FLIR Image that results after the acquisition is processed in parallel by the pedestrian detector and the semantic road segmentation module. The results of the pedestrian detection are forwarded to the pedestrian tracking and distance estimator that output features like pedestrian motion direction, speed and distance with respect to the car. All these features and the position of the pedestrian with respect to the road are input to the action recognition module that predicts the cross or not cross action using a time series Long Short Term Memory model.

A. PEDESTRIAN DETECTION AND TRACKING MODULES

A YOLO [18] type architecture with spatial pyramid pooling was adopted for detecting pedestrians in infrared images. This choice is made due to the high classification accuracy obtained with such a network in previous work [19]. The algorithm employed by YOLO splits the image into multiple regions in which weighted bounding box predictions are made. The weights are obtained using bounding box priors. These are computed by K-means clustering on the input training dataset.

The proposed tracking method, which will be discussed in detail in this section, consists of the following major components: data association and similarity cost computation, track selection, update and refinement. The pedestrian tracking algorithm using a far infrared camera is one of the contributions of this paper.

We build upon the state of the art by creating a loosely coupled tracking solution that follows the track by detection framework and we engineer a similarity cost that includes both motion and appearance scores thus making better correspondences between the tracks and detections. We make an optimal assignment between tracks and detections using an optimization algorithm and finally we refine the results removing any unwanted tracks. The input to our algorithm is a set of bounding boxes corresponding to the detected objects, which also have the classification probability for that object class. The output is given by the set of tracked objects that have a unique ID associated to them and a smoothed trajectory.

1) Data Association and Dissimilarity Cost Computation

In the presence of clutter, it is often difficult to distinguish sensor measurements from false alarms. Furthermore, computing an association score between a track and every detection in a frame is a computationally intensive task. Hence, a measurement validation gate is used to reduce the number of comparisons by forming a gate around the position of the predicted hypothesis and only considering detections within that region. The gate is described by an origin (which is usually the position of the predicted value $\overline{X}_k$) and a gate volume V_k. The validation region for ellipsoidal gating is given by equation (1).

$$(\chi_k^i - \overline{X}_k)' S_k^{-1} (\chi_k^i - \overline{X}_k) \leq \gamma \tag{1}$$

In equation (1) χ_k^i is the i^{th} measurement inside the validation gate S_k which is defined in [31], and it represents the innovation co-variance, while γ is a probability threshold which can be obtained from tables of the chi-squared distribution and it is kept constant for a given application. The gate volume is given by equation 2, where c is a scaling value.

$$V_k = c\gamma^{\frac{1}{2}} |S_k|^{\frac{1}{2}} \tag{2}$$

A graphical depiction of the grating process can be seen in Figure 6.

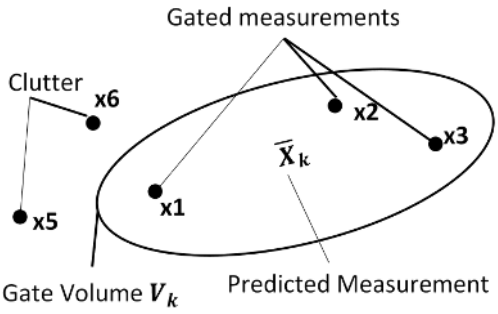


FIGURE 6: Graphical depiction of the measurement validation gate

Some far infrared cameras (including ours) can have freezing moments in which frames are not acquired for a number of seconds. Due to this phenomena the tracked objects may be at larger distances than predicted (due to the loss of measurements). The proposed model was build to cope with such situations by inferring two values for the parameter c. The far infrared camera signals the freezing moment by displaying a small white square in the bottom right corner,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

hence we know when to apply a more reasonable value for the variable c. The expression for c when the camera is functioning normally is displayed in (3) and for the frames following a freezing moment the expression is depicted in (4).

$$c = 2 \times (w + h) \tag{3}$$

$$c = \frac{2wh}{3} \tag{4}$$

The scaling of the gate volume depends on the tracked object dimension, w represents object width and h represents object height. The two expressions for calculating the value c were determined experimentally. After obtaining the gated measurements for a track, a similarity cost is computed between the track and all measurements within the validation gate.

In the proposed solution, the similarity cost $\epsilon(i, j)$ (5), that is computed between the i^{th} infrared measurement and the j^{th} object in the tracking list, is defined as the sum of two distance measures, one representing a motion score $m(i, j)$ and another representing an appearance score $a(i, j)$. Each of the two scores is a weighted sum of several terms which will be described shortly.

$$\epsilon(i, j) = a(i, j) + m(i, j) \tag{5}$$

One of the main difficulties when building similarity cost functions is trying to solve some problems without unsolving others. To this end each time a new term was introduced in the cost function equation with the purpose of solving an issue, all the scenarios corresponding to the already available terms were tested as well to ensure they are still working.

In the proposed solution we have decided to engineer the cost function because such a solution would offer more control regarding the effect of each feature that is used in the cost computation, so the final equation is not a black box. Furthermore, we know which feature is responsible for solving certain issues. Solutions based on neural networks would not give us the flexibility mentioned above, and in case a scenario is presented to the network that was not covered by the training test, the network might fail or latch onto the wrong object.

Appearance Score

The appearance score is important in tracking because it can offer a way to recognize an object in different frames and is also a measure of distinguishing between different objects when they are in proximity of each other. Nonetheless, the appearance of an object may be altered in consecutive frames due to deformations or changes of view point. The thermal infrared emission is independent of any light source, however the combination between the human skin infrared emissivity and the clothes that each person wears, leads to a unique thermal signature for each subject. Therefore, it is important to define a method that captures the changes in appearance and the texture uniqueness of each pedestrian. To address this problem in this work, we design an appearance score that relies on multiple weighted features. The expression of the appearance score between the tracked object and the infrared detection is given in (6).

$$\begin{aligned} a(i, j) = w_{huLbp} \times huLbp(i, j) + w_{\mu s} \times \mu s(i, j) + \\ + w_{\sigma s} \times \sigma s(i, j) + w_{hs} \times hs(i, j) + \\ + w_{Ws} \times Ws(i, j) + w_{cs} \times cs(i, j) + \\ + w_{os} \times os(i, j) \end{aligned} \tag{6}$$

The terms $w_{huLbp}, w_{\mu s}, w_{\sigma s}, w_{hs}, w_{Ws}, w_{cs}, w_{os}$ are the weight contributions for each distance measure. They were determined experimentally by evaluating each term's contribution over 160 sequences recorded with the thermal camera in different conditions including day, night, cold and warm scenarios. The values determined experimentally for each weight are $w_{huLbp} = 10, w_{\mu s} = 285, w_{\sigma s} = 8, w_{hs} = 10, w_{Ws} = 10, w_{cs} = 550, w_{os} = 95$. The values determined for the weights are not unique and small variations are possible without affecting the output of the algorithm. For readability, some weights were approximated to the nearest multiple of 5, where it was possible.

The meaning of each distance measure from the appearance cost equation is the following: $huLbp(i, j)$ represents the histogram of uniform local binary pattern (LBP) in the region of interest (ROI) given by the detection, $\mu s(i, j)$ is the mean value pixel intensity distance of the ROI, $\sigma s(i, j)$ represents the variance score in the ROI, $hs(i, j)$ and $Ws(i, j)$ are the height and width distances, $os(i, j)$ represents the overlapping score and $cs(i, j)$ represents the class detection probability score.

To capture the texture structure of each hypothesis, in order to use it in the track and measurement association, we have used a uniform local binary pattern histogram over the region of interest. The object level structure can be a good feature to measure the correlation between a track and a measurement in adjacent frames, due to the fact that the structure of an object is not expected to change drastically in consecutive frames. The LBP descriptor outputs a binary word for each pixel as shown in (7):

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \times 2^p \tag{7}$$

The number of neighbors to be analyzed on a circle of radius R is given by P, $s(x) = 0$ if $x \geq 0 \wedge s(x) = 1$ otherwise; g_p is the intensity of neighbor p and g_c is the intensity of the center pixel. In the proposed solution a 3×3 neighborhood is used. All LBP codes from the region of interest can be represented in the form of a 256-bin histogram. In order to achieve a faster feature comparison, reduce the memory consumption and achieve more robustness to noise, a uniform local binary pattern histogram is employed. A LBP is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 and vice versa when the bit pattern is traversed circularly [51]. Therefore, by comparing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

pixel values in a 3×3 neighborhood, there are a total of 256 patterns, 58 of which are uniform, which yields in 59 different labels. The voting of each LBP code in the uniform LBP histogram is done via a look up table in order to improve the running time efficiency of the feature extraction. The intuitive depiction of the uniform LBP histogram creation is depicted in Figure 7.

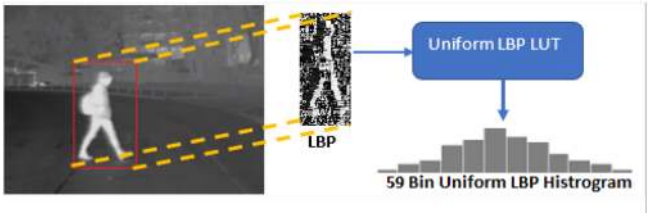


FIGURE 7: Computation of the LBP descriptor for the region of interest, and the creation of the uniform LBP histogram using a Uniform LBP LUT.

The final value of the histogram of uniform local binary patterns, $huLbp(i,j)$, term is obtained by performing a root mean square operation on uniform LBP histograms contained by the measurement j, $huLbp(j)$, and the one stored by track i, $huLbp(i)$ as illustrated in (8).

$$huLbp(i,j) = \sqrt{\frac{1}{59}\sum_{k=1}^{59}(huLbp(i)_k - huLbp(j)_k)^2} \quad (8)$$

To compute some of the dissimilarity values from the appearance score the function expressed in (9) is defined.The operator $|a|$ refers to the absolute value of the variable a.

$$F(x,y) = |x-y| \quad (9)$$

The mean and standard deviation of the region of interest are two measures that are used in the appearance score. Mainly due to the fact that the thermal camera does not need an external source of illumination the two mentioned values do not have large variations between consecutive frames for the same object instance (because the pedestrian cannot increase his temperature abruptly in consecutive frames). The mean is a measure of the intensity, while the standard deviation is a measure of the contrast in the region of interest, both characterize the level of thermal infrared radiation emitted by the pedestrian. The scores of $(\mu s(i,j))$ is obtained by applying (9) as shown in equation (10).

$$\mu s(i,j) = F(\mu s(i), \mu s(j)) \quad (10)$$

The value of $(\sigma s(i,j)$ is computed analogously by applying the function F defined in (9) with the parameters $(\sigma s(i))$ and $(\sigma s(j))$. The value of $\mu s(i)$ and $\sigma s(i)$ is computed as illustrated in (11) and (12), where h and w are the dimensions of the regions of interest and $Image$ is the region of interest from the far infrared image.

$$\mu s(i) = \frac{1}{M}\left(\sum_{k=0}^{h-1}\sum_{r=0}^{w-1} Image(k,r)\right) \quad (11)$$

$$\sigma s(i) = \sqrt{\frac{1}{M}\left(\sum_{k=0}^{h-1}\sum_{r=0}^{w-1}(Image(k,r) - \mu s(i))^2\right)} \quad (12)$$

The physical attributes of each pedestrian, like width and height, are other properties that offer clues when performing object association. Physical particularities of the detected pedestrians do not change suddenly and due to the fact that the frame rate of the used thermal sensor is sufficiently high, we can capture the variations for the same object instance. Even though properties such as width and height can help distinguish between pedestrians of different noticeable sizes, in case the pedestrians are similar in dimensions these properties alone are not sufficient. For this reason, the mentioned features are introduced in the appearance score among other functions. The height measure for track j and detection i, $hs(i,j)$, is computed by applying the function F(h(i),h(j)) defined in (9), where $h(x)$ is the height of instance x. The width distance score, $Ws(i,j)$, is computed analogously taking the width measure instead of the height.

In the aggregate cost function the classification probability coming from the pedestrian classifier in thermal images is also included. It was noticed that the classification score difference from adjacent frames for the same pedestrian instance is very small compared to the difference obtained by subtracting the classification score for different pedestrian instances from consecutive frames. The classification score $cs(i,j)$ is obtained by applying function F(c(i),c(j)) (9), where the c(x) represents the classification score of x. The result is a value between 0 and 1, the closer the difference is to 0 the more similar the two objects are with respect to this distance metric. It may happen that multiple pedestrians have similar classification scores, in such a scenario, this metric is not sufficient to discriminate between objects, hence the value was used as a component of the appearance cost not just by itself.

The last term of the appearance cost function is a size-based distance function. This term incorporates the size similarities of a track and a measurement, as well as localization information of each detection compared to the predicted localization information from a track. Therefore, the size-based distance, $os(i,j)$, between a measurement i and track j is considering both the location and size of the bounding boxes and is defined in equation (13); where A_i is the area of the measurement, A_j is the area of the tracked object, and $A_\cap$ is the area of the intersection between the two objects.

$$os(i,j) = \frac{|A_j - A_i|}{A_\cap} \quad (13)$$

Motion Score

In some situations when two pedestrians are very similar as viewed from the thermal camera, the appearance score might

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

be unable to distinguish between them. For such scenarios, the motion pattern of each pedestrian is also included in the final cost. The final expression for the motion score, $m(i,j)$, between measurement i and track j is shown in (14). As in the case of the appearance score, the weights were determined experimentally, their values are $w_{dst} = 85$, $w_{\sigma m} = 20$.

$$m(i,j) = w_{dst} * dst(i,j) + fc(i,j) + \\ + w_{\sigma m}(\sigma m(i,j)_x + \sigma m(i,j)_y) \quad (14)$$

When tracking an object in adjacent frames, its motion is offering an important clue regarding the objects future position in the next frame. We define the difference between the predicted position and the measured position as the motion-based distance measure. The Euclidean norm is used, and the center positions of the objects bounding boxes expressed in 2D image coordinates are selected when computing the distance.

Another term that is included in the motion score is the flow distance metric between the detection i and track j. Tracking using a sparse optical flow algorithm may not be reliable from a qualitatively point of view and dense optical flow is a very expensive procedure computationally and qualitatively may be imperfect on unstructured surfaces. Although the individual trajectories that result from the optical flow may be inaccurate, collectively they can provide clues regarding the motion of objects in consecutive frames. After applying the algorithm presented in [52] for computing the optical flow, several steps were performed for obtaining the angle and magnitude values for the optical flow of the region of interest. First of all, 36 bins are created to store the flow values. Each flow vector value casts a vote in one of the 36 bins based on its angle. Secondly, after having all the identified flow vectors vote inside their corresponding bins, we compute the mean values for the magnitude and angle for each bin. Finally, a search is performed to find the bin where the majority of votes were cast and afterwards the mean magnitude and angle corresponding to that bin are selected as flow parameters for our region of interest. We have observed using multiple sequences recorded in various scenarios, that objects do not change their motion pattern abruptly in consecutive frames, a thing which has led us to define the flow cost as shown in 15.

$$fc(i,j) = F(\theta_i, \theta_j) \times w_\theta + F(\vartheta_i, \vartheta_j) \times w_\vartheta \quad (15)$$

The flow angle is represented by (θ), the flow magnitude is represented by (ϑ), the function F has been defined in (9) and $w_\theta = 40$ and $w_\vartheta = 15$ are two weights whose values were determined experimentally.

The last term of the motion score, σ_m, represents the deviation of object motion from the objects current motion pattern, on x and y directions. We have included this term because we want to penalize large deviation from the current motion pattern of the object. The rationale behind this term is: if we consider the motion pattern of the pedestrian in the last five frames, the next move will most likely resemble the same pattern, having a small deviation for the correct object association. We would also want to mention we stored the last five positions for each tracked object. The expression for variation cost distance is displayed in equation (16). This cost is applied on both x and y components of the 2D motion. The variable X_i represents a detection, $Z_j(k)$ represents the kth stored past position of track j.

$$\sigma m(i,j) = |X_i - \sqrt{\frac{1}{5}\sum_{k=0}^{4}(Z_j(k) - Z_j(k+1))^2}| \quad (16)$$

It is worth mentioning the fact that the identified weights for the appearance and motions scores are not unique, other variations are possible however the selected values offered the best results in our case.

Track Selection, Update and Refinement

Once the motion and appearance similarity scores have been computed between measurement i and track j, they can be assembled into the final cost $\epsilon(i,j)$ as shown in equation (5).

The similarity cost is computed for all tracks stored in memory against all the detections from the current frame that fall within the track co-variance ellipse. The affinity scores are stored into a matrix format and are used as input in the Hungarian [53] algorithm, that finds the best assignment for each detection in the current frame with the corresponding track. If the similarity cost for a track-detection assignment pair is above a threshold, the assignment is nullified and a new track is created for that measurement.

After finding all the viable correspondences between the tracks and measurements, the following scenarios can be identified: we can have a track matched with a detection, an unmatched detection or an unmatched track. In the case of a successful track and measurement association, the track and all its parameters are updated, using the new information coming from the measurement. In case we have an unmatched detection, a new track is created. The newly created track will remain in an unstable state until it will be associated to new detections and tracked for another five frames, and afterwards it will become stable and it will be displayed.

One of the key features of a tracking algorithm is to maintain the tracked object even if the detection is not available for a number of frames due to errors in the object detector, occlusions and other factors. For this reason, each track incorporates a history counter, which counts the number of frames for which a specific track has not been associated. The position of the unmatched track in the next frame is predicted based on the motion pattern the track has had so far, using a Kalman filter predict function. After a number of frames, if the unmatched track remains un-associated it enters a drifting state, where the track is not displayed however it is still kept in memory. The track is finally removed in the drifting stage if still not matched. Therefore, unmatched tracks are not removed immediately. It is important to mention the fact that tracks that exit the region of interest are marked

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

for termination and removed. In the proposed solution the track history and drifting history have different values for day and night scenarios, which were determined experimentally and depend on the camera frame rate. The value for the history counter is 20 and for drifting counter is 15 for night scenarios, while for day cases history counter becomes 25 and drifting counter becomes 15. It is worth mentioning the fact that the drifting and history counters values mentioned above are applicable only to stable tracks. In case of tracks which are not stable, they will be set for removal if they are not associated after 5 frames for night scenarios or 7 frames for day scenarios. In Figure 8 a scenario is illustrated where a pedestrian gets occluded by some trees, but his identity is maintained until he becomes visible again and the object detector is once again able to successfully detect him.

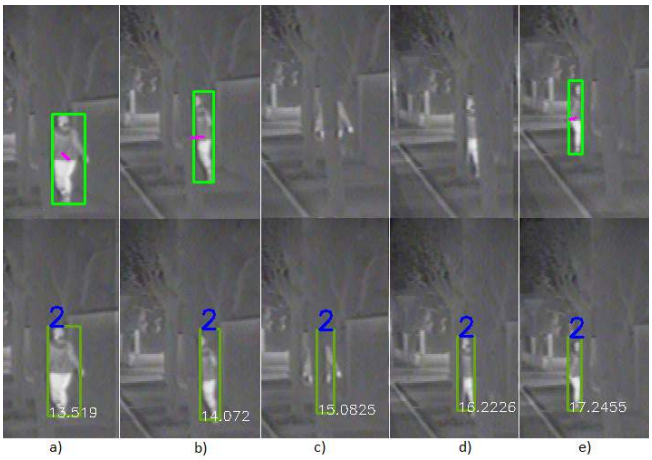


FIGURE 8: a)Pedestrian fully detected and tracked b)pedestrian begins to get occluded; c) pedestrian is fully occluded but continues to be tracked; d) pedestrian is occluded we can observed parts of him between the trees, the detector is unable to detect him, but due to the tracking algorithm his identity is maintained; e) Pedestrian reappears and is detected and tracked.

Finally, the track list is updated with the newly found tracks and old tracks are removed. In Figure 9 we show a scenario where two pedestrians cross paths. The proposed solution is able to maintain the correct identity of each pedestrian and not latch onto the wrong pedestrian when the pedestrians overlap. The bottom right image from Figure 9 shows the path history of each pedestrian position.

Another scenario where multiple pedestrians walking on a sidewalk are being successfully tracked is displayed in Figure 10. The pedestrian ID is maintained and there is no ID switch error for the pedestrians that are close to each other. The meaning of the four images that make up Figure 10 remain the same as in the case of Figure 9.

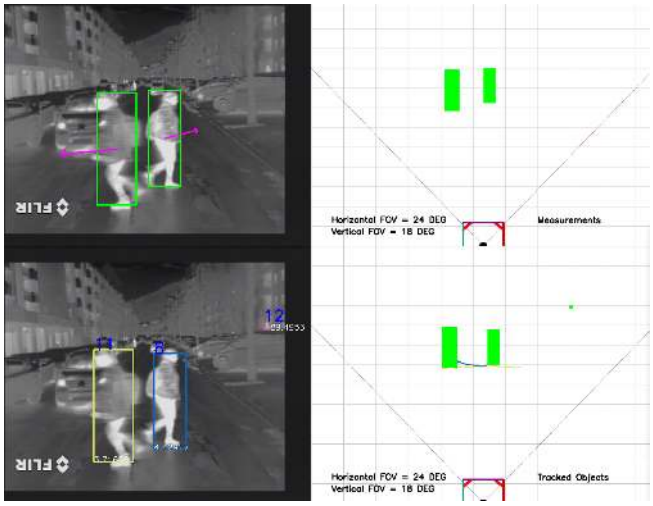


FIGURE 9: In the top left, object detections are shown, with their corresponding motion vectors. In the top right, the detections are projected in a grid. In the bottom left each track is represented with a unique id and color and in the bottom right the tracked objects are depicted as well as the motion trail corresponding to the path of each pedestrian.

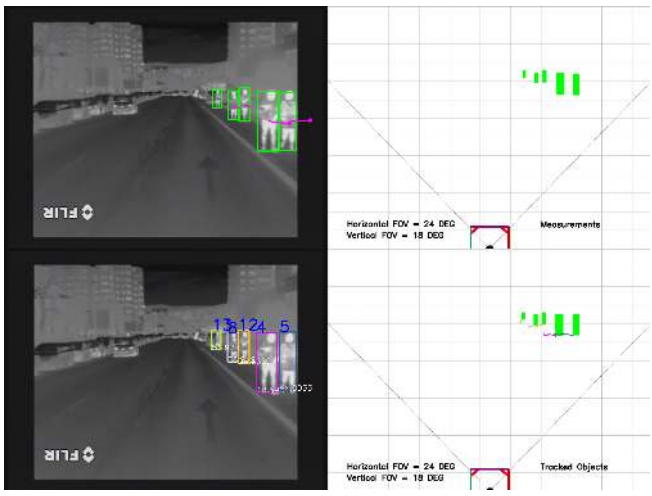


FIGURE 10: In the top left, the measurements are shown. In the top right, the image illustrates the detections projected onto a grid. In the bottom left the tracked objects are shown, and in the bottom right the trail left by the tracked object is displayed.

B. SEMANTIC SEGMENTATION OF THE ROAD SURFACE IN INFRARED IMAGES

A key factor that influences the decisions of a pedestrian action recognizer is given by the position of the pedestrian with respect to the street, or with respect to the drivable area. A semantic segmentation of the road in infrared images is embraced in this paper. The convolutional neural network model proposed by [54], [55] is adopted. The network contains a sequential architecture based on an encoder segment producing downsampled feature maps and a subsequent decoder

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

segment that upsamples the feature maps to match input resolution. We have kept the original architecture proposed by [54], [54] which consists in 16 layers that combine residual blocks and downsampling blocks defining the encoder, while layers 17 to 23 form the decoder. The decoder includes transposed convolutions that have the role to upsample the encoder's feature maps. The result of the segmentation is a labeled image of size equal to the input image size having the pixels labeled as either road or non-road.

In our experiments we have used the PyTorch implementation and the pretraied encoder provided by ERFNet. This encoder was pre-trained on ImageNet while the decoder was trained from scratch on the infrared images. We have modified the number of classes to two (road and non road pixels), and trained the network with a batch size of 6 for 150 epochs.

C. PEDESTRIAN CROSS ACTION RECOGNITION

The cross action recognition module represents another original contribution of this paper as it engineers a times series prediction model for action recognition in infrared images. The input of this module consists in time series feature vectors of maximum length equal to t and the output is a cross action recognition probability vector for frame t. As time series prediction model we use the classical Long Short Term Memory (LSTM) Network proposed by [56] and extended by [57], [58]. It is able to process sequential data one sample at a time and its additive interactions improve the gradient flow through the network.

The LSTM network employed in this paper is a many-to-one topology for classification. Its structure is presented in Figure 11. The input is formed of feature vectors computed

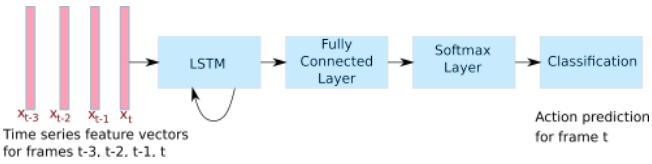


FIGURE 11: LSTM network topology: considers as input the time series feature vectors for frames $t-3$, $t-2$, $t-1$, t and predicts the action for frame t

for continuous frames in the video sequence. The feature vectors are detailed in section IV-C. To predict class labels, the network ends with a fully connected layer, a soft-max layer, and a classification output layer.

The LSTM layer in Figure 11 contains several LSTM cells [56], [59]. Each cell comprises computational blocks, named gates, that control the amount of information that is added or removed by the cell. The LSTM layer with three cells, from Figure 12 depicts the data flow from frame $t-2$ to frame t.

The input vectors for timestamp t, $t-1$, $t-2$ and $t-3$ are $x_t, x_{t-1}, x_{t-2}, x_{t-3}$. Two states, h_t –the hidden state and c_t – the cell state are maintained at each time stamp. The i, f, g, and o represent the input gate, forget gate, cell candidate, and output gate. The input gate i is beneficial for storing new

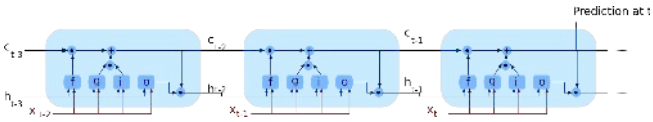


FIGURE 12: Three LSTM cells and the data flow from frame $t-2$ to frame $t-1$ and to frame t as implemented by [59]

information in the cell, the forget gate, f helps in discarding/forgetting irrelevant information from the previous state, the cell candidate gate g is used for updating the cell state and the output gate, o controls which information is transmitted to the next time step. The cell state at a given time step t is given by:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{17}$$

where $\odot$ represents the element-wise multiplication of vectors. The hidden state at time step t is:

$$h_t = o_t \odot \tanh c_t \tag{18}$$

As described by [58] and [59] the weights that are learned and updated during the training process of an LSTM are the input weights W, the recurrent weights R, and the bias B:

$$\begin{aligned} W &= \begin{bmatrix} W_i & W_f & W_g & W_o \end{bmatrix}^T, \\ R &= \begin{bmatrix} R_i & R_f & R_g & R_o \end{bmatrix}^T, \\ B &= \begin{bmatrix} B_i & B_f & B_g & B_o \end{bmatrix}^T, \end{aligned} \tag{19}$$

At time step t the behavior of the gates in the cell is defined as follows:

$$\begin{aligned} i_t &= \sigma_g(W_i x_t + R_i h_{t-1} + B_i) \\ f_t &= \sigma_g(W_f x_t + R_f h_{t-1} + B_f) \\ g_t &= \tanh(W_g x_t + R_g h_{t-1} + B_g) \\ o_t &= \sigma_g(W_o x_t + R_o h_{t-1} + B_o) \end{aligned} \tag{20}$$

where σ_g is the sigmoid function, $\sigma(x) = (e^{-x} + 1)^{-1}$. In the implementation for this paper the LSTM network functionality provided by [59] was used. The fully connected layer is used to combine the features in order to classify the actions. The output size of the fully connected layer is equal to two, as we have two actions which are to be recognized. The softmax layer applies a softmax function to the output of the fully connected layer. This layer is followed by the classification layer that has the role of computing the cross entropy loss during the network training procedure.

Features used by the LSTM

The cross action recognition is based on a time series analysis of the pedestrian's position in the image, motion features, distance of the pedestrian with respect to the ego-vehicle, and road context information as shown in Figure 13.

The feature vector for a pedestrian track k in frame t contains:

- Bounding box parameters of the tracked pedestrian: $BB_{kt} = \left[x_{kt}^{top} y_{kt}^{top}, w_{kt}, h_{kt}\right]$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

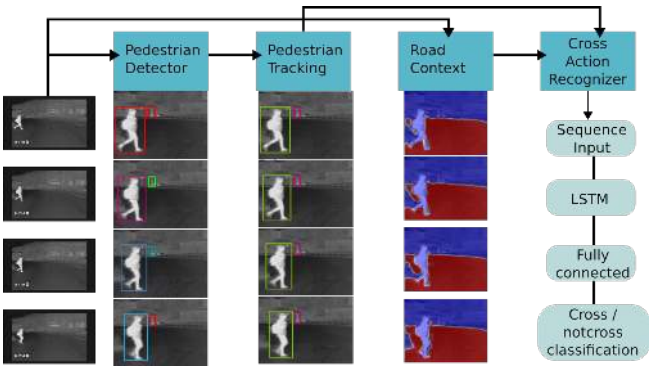


FIGURE 13: Features computed for the time series analysis– from left to right we consider the bounding box information provided by the pedestrian detector, the pedestrian speed provided by the tracking module, the pedestrian relation with respect to the road and the distance of the pedestrian with respect to the ego-vehicle.

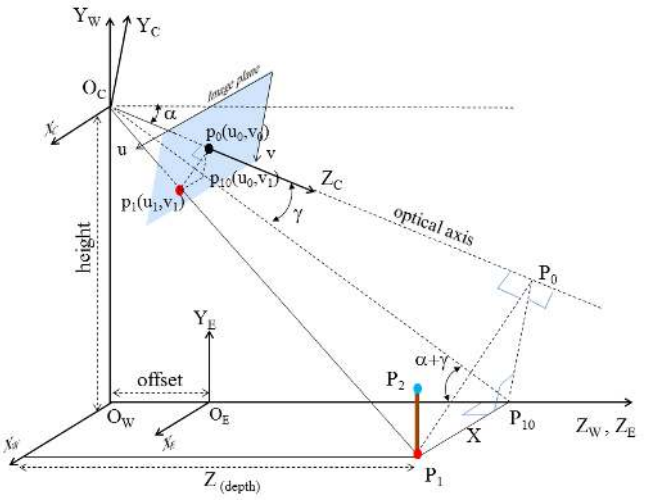


FIGURE 14: Initial setup of the scene geometry: $(O_C X_C Y_C)$ denotes the camera coordinate system, with blue we have the image plane and $p_1(u_1, v_1)$ is the projection on the image plane of a 3D point P_1 situated on the road plane

- The horizontal and vertical optical flow components, $O_{kt} = [o^x_{kt}, o^y_{kt}]$
- Horizontal speed of the tracked pedestrian: s^h_{kt}
- Distance from the ego-vehicle to the pedestrian: z_{kt}
- Road context feature vector R_{kt}.

The bounding box parameters are computed by the pedestrian detection and tracking module. The horizontal and vertical optical flow components are computed in the feature extraction phase of the tracking module. In our experiments a monocular infrared camera was used for acquiring the sequences, hence the distance estimation method from the monocular camera was implemented. The top view projection of the scene was also used for computing the relative speed on the horizontal direction (Ox). The speed was temporally filtered by a low pass average filter of dimension 5. The horizontal speed is computed based on tracking and 3D information, while the road context feature vector computation is explained bellow.

In order to estimate the pedestrian distance with respect to the ego-vehicle an approximate method for distance measurement using a monocular camera is employed. The constraints considered in the proposed approach are shown in Figure 14.

As shown in Figure 14 we consider the following assumptions in order to compute distance relative to the ego-vechicle coordinate system :

- Consider 3 coordinate systems: the Ego-vehicle coordinate system $(O_E X_E Y_E Z_E)$, the Camera coordinate system $(O_C X_C Y_C Z_C)$ and the World coordinate system $(O_W X_W Y_W Z_E)$, which is related to the road (considered flat);
- Their relative position and orientation is established during the system set up as presented in Figure 14: between the ego-vehicle and the world coordinate systems there is only a translation $(offset)$ along the Z direction (due to the camera mounting system on the ego-vehicle); between the world coordinate system and the camera coordinate system there is only a translation along the Y axis (camera mounting $height$ above the ground) and a rotation around the X axis (pitch angle α)
- Let O_w be the projection of the camera's optical center (O_c) on the road plane. We consider O_w to be the origin of the world coordinate system in which all the 3D measurements are computed. The transformation of the 3D coordinates from the world into the the ego-car coordinate system can be done by a simple translation along the Z axis by subtracting the $(offset)$;
- The extrinsic parameters of the camera model $(offset$, $height$ and $\alpha)$ can be precisely estimated during the system setup. $Offset$ and $height$ are measured using a laser rangefinder while the pitch angle (α) is computed as $\alpha = tan^{-1}(O_w O_C / O_w O_{P10})$ applied on the $O_c O_w O_{P10}$ triangle (Figure 15) where O_{P10} is the intersection of the optical axis with the road plane. In order to determine the 3D coordinate (mainly the depth $O_w O_{P10}$) of the point O_{P10} a white cross was drawn on the image, centered in the principal point (Figure 21) and a corresponding marker was drawn on the road surface in such a manner that its image projection perfectly overlapped the white cross. The measured depth of the marker relative to point O_w is the length of the $O_w O_{P10}$ segment;
- We also know the camera intrinsic parameters [60], from the camera dustsheet:
  - f_x - focal length measured as number of horizontal pixels
  - f_y - focal length measured as number of vertical pixels
  - $p_0(u_0, v_0)$ - the principal point measured in pixels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

The focal length expressed in pixels ($f[pixels] = f[mm]/PixelSize[mm]$ according to [60]) is used to transform the pixel coordinates into metric units as it will be shown in equations (21) - (27).

Let suppose that we observe a 3D vertical segment (i.e. the median vertical axis of the pedestrian - Figure 14) having as extremities the 3D points $P_1(X, 0, Z)$ and $P_2(X, Y, Z)$. Their projections on the image plane are the 2D points $p_1(u_1, v_1)$ and $p_2(u_2, v_2)$ The goal is to compute the 3D coordinates of the points P_1 and P_2 in the world coordinate system and by translation in the ego-vehicle coordinate system.

First a side-view projection of the scene on the $Y_wO_wZ_w$ plane (having $X_w = 0$) is performed as depicted in Figure 15. Point P_1 is projected in point P_{10}, p_1 is projected in p_{10} and P_2 in P_{20} and so on.

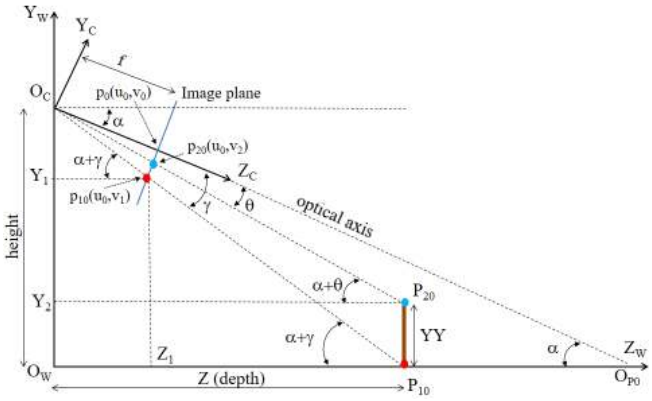


FIGURE 15: The side-view projection of the points in the scene

Based on trigonometric relations in the triangle $O_cp_{10}p_0$ one can compute the angle γ:

$$\gamma = tan^{-1}\left(\frac{v_1 - v_0}{f_y}\right) \tag{21}$$

Using the relations in triangle $O_cO_wP_{10}$ the depth of point P_{10} (respectively P_1 in the world coordinate system) can be deduced as:

$$Z = [O_wP_{10}] = \frac{height}{\tan(\alpha + \gamma)} \tag{22}$$

From the right triangle $O_cp_{20}p_0$ we can compute the angle θ which will be used further for the height computation of the object/pedestrian YY:

$$\theta = tan^{-1}\left(\frac{v_2 - v_0}{f_y}\right) \tag{23}$$

The height YY is computed from the right triangle $O_cY_2P_{20}$:

$$YY = [O_cO_w] - [O_cY_2] = height - Z \cdot \tan(\alpha + \theta) \tag{24}$$

The top-view or bird-eye view projection of the scene on the horizontal road plane $X_wO_wZ_w$ (Figure 16) is used to compute the X coordinate – the lateral offset of point P_1 with respect to the axis O_wZ_w:

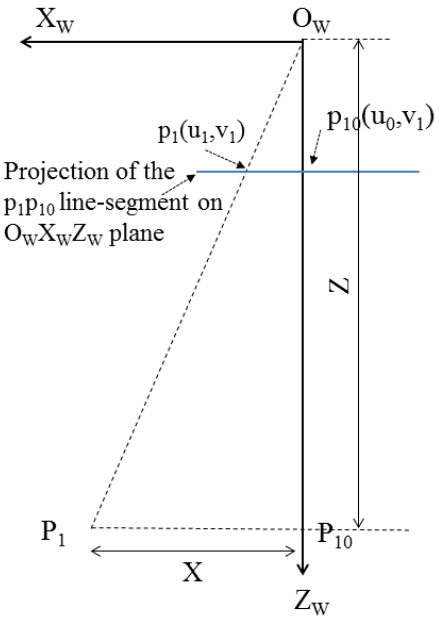


FIGURE 16: The top-view projection of the scene on the horizontal plane $X_wO_wZ_w$

$$X = [P_1P_{10}] = [p_1p_{10}] \cdot \frac{[O_wP_{10}]}{[O_wZ_1]} = \frac{(u_1 - u_0)}{[O_wZ_1]} \cdot Z \tag{25}$$

The length of the segment $[O_wZ_1]$ where Z_1 is the projection of the point p_{10} on the horizontal plane can be deduced from the side view projection:

$$[O_wZ_1] = [Y_1p_{10}] = [O_cp_{10}] \cdot \cos(\alpha + \gamma) = \frac{f_y}{cos(\gamma)} \cdot \cos(\alpha + \gamma). \tag{26}$$

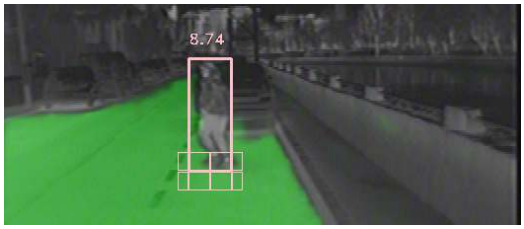
From equations (25) and (26) the the X coordinate is computed:

$$X = \frac{u_1 - u_0}{f_y} \cdot \frac{Z}{\cos(\alpha + \gamma)} \cdot \cos\gamma \tag{27}$$

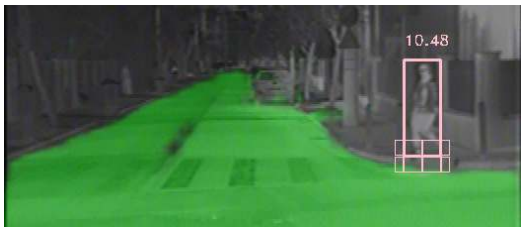
In our experiments a monocular infrared camera was used for acquiring the sequences, hence the distance estimation method from the monocular camera was implemented. The top view projection was also used for computing the relative speed on the horizontal direction (Ox). The speed was temporally filtered by a low pass average filter of dimension 5.

The road context features are computed inside a set of 8 rectangles at the bottom of the bounding box, because that is the place having a high probability of the pedestrian touching the ground (when lower body occlusions are not present) - see Figure 17 and Figure 18. The number above the bounding box in Figure 17 represents the distance in meters between the pedestrian and the ego-vehicle computed with the distance estimation algorithm above, while the green part of the image represents the result of the road segmentation module.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access



(a) Pedestrian on street



(b) Pedestrian outside the street

FIGURE 17: Rectangles in which road context features are considered: (a) the pedestrian is on the road, (b) the pedestrian is outside the road, but very close to it. Road pixels are marked with green.

The eight rectangles have a dimension proportional with the size of the pedestrian bounding box. In each of these eight rectangles the average number of road pixels is computed. If a pedestrian is on the road, these average values will be high in most of rectangles, exceptions being made by the two rectangles that usually capture the feet, that will have a lower street pixel average. If the pedestrian is outside the road these averages will be low. The size of the eight rectangles is based on the width and height of the bounding box as shown in Figure 18. The dimension of the eight rectangles are either

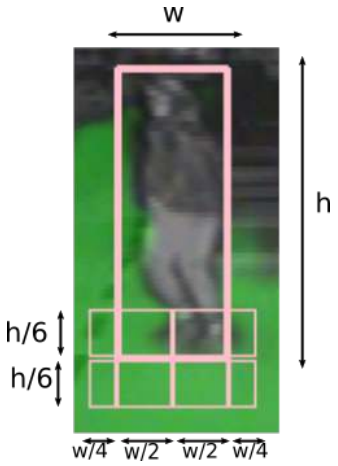


FIGURE 18: The rectangles in which road features are considered: for each of the eight rectangles in the lower part of the pedestrian, the average number of road pixels is computed.

equal to $\frac{h}{6} \times \frac{w}{4}$ or to $\frac{h}{6} \times \frac{w}{2}$, where h is the height of the bounding box and w is its width (both expressed in pixels). These sizes have been chosen empirically.

V. EXPERIMENTS AND RESULTS

The proposed solution is based on a tight combination of modules that provide the features for the action recognition model. Each module was trained and evaluated separately. The evaluation metrics and results for each module are described in what follows.

A. PEDESTRIAN DETECTION AND TRACKING

The YOLO based pedestrian detector was trained on FLIR-ADAS [48] dataset and fine tuned for the CROSSIR dataset. Starting with the weights of the FLIR-ADAS model obtained by [19] for our experiments we have trained YOLO on the annotated pedestrians in our dataset. The training was done for 20000 iterations. The model with highest mean average precision is kept. We only consider pedestrian training samples with a minimum bounding box width of 30 pixels and with no occlusions.

To compare the performance of the proposed far infrared tracker with other state of the art solutions, we have used the PTB-TIR benchmark dataset. In this dataset there are multiple thermal image sequences each having manual annotations.The center location error (CLE) is an average euclidean distance between the tracked object position and the ground truth position for that object. If the CLE is within a given threshold (20 pixels on the PTB-TIR benchmark) the tracking is said to be successful at this frame. The precision score measures the percentage of how successful is the tracking on the data-set. Apart from the dataset, evaluation results from multiple types of trackers on the given sequences are available such that the strengths and weaknesses of each solution can be observed comparatively. In the evaluation of the proposed tracking solution on the PTB-TIR benchmark, we have selected to include only the sequences that are related to pedestrians as seen from a vehicle mounted thermal sensor (since our tracker has been specifically tailored to track pedestrians for the field of intelligent vehicles). The results of the proposed solution on the benchmark are displayed in Figure 19, under the name OURS along with other representative approaches from the literature. It is worth mentioning that the proposed solution comprises no hardware acceleration methods and the data association function was engineered. Consequently we were able to monitor the impact of each feature independently, while keeping a clear view on the feature extraction part from the data association module. The numeric results from Figure 19 are also displayed in Table 1.

B. ROAD SEGMENTATION

The training of ERFNet [55] was done using the Adam optimizer [74], with a batch size of 6, momentum of 0.9, weight decay of $2e^{-4}$ and a starting learning rate of $5e^{-4}$. The learning rate is set every epoch according to the formula below:

$$lr_i = \left(\frac{1 - (i-1)}{m} \right)^{0.9} \tag{28}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

Precision Results	
Method	Tracking Precision Score
ECO [61]	89.8%
SRDCF [62]	88.8%
DeepSTRCF [63]	88.8%
OURS	85.6%
Staple [64]	83.6%
MLSSNet [65]	83.4%
CFNet [66]	83.3%
VITAL [67]	82.8%
ECO-HC [68]	82.4%
TADT [69]	77.9 %
MCFTS [70]	77.6 %
HDT [71]	75.8%
HCF [72]	75.6%
SiamFC [73]	74.5 %

TABLE 1: Comparative evaluation for various tracking solutions

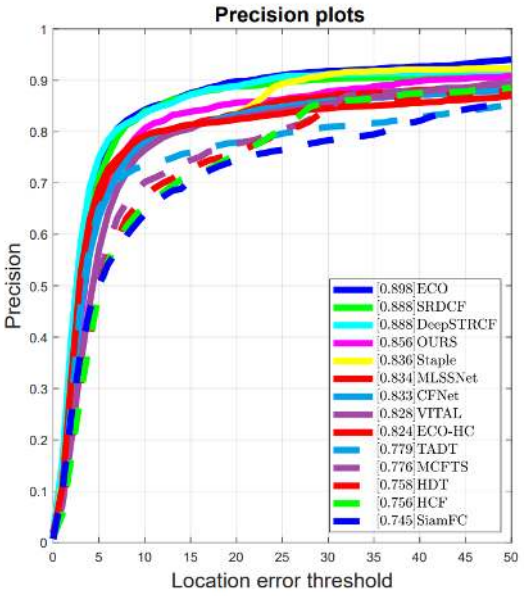


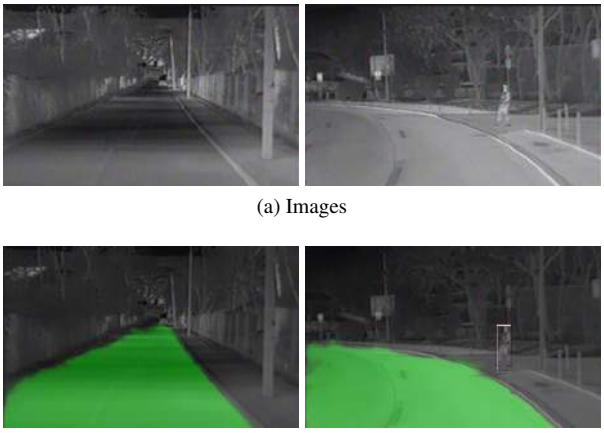
FIGURE 19: Comparative evaluation of the tracking solution with respect to the precision of the solution

where i is the current epoch number and m is the total number of epochs over which the model is trained. Convergence is achieved after 150 iterations with a segmentation accuracy of 83.78% obtained on the test set. The training set contains 500 labeled images, the validation set contains 147 images and the test set contains 200 images.

Figure 20 shows some segmentation results. The original images are shown in the top part of the figure, while the bottom row of Figure 20 displays in green the pixels having a high probability to belong to the road.

C. DISTANCE ESTIMATOR EVALUATION

For the quantitative assessment of the proposed monocular distances estimator, firstly a static scenario was considered (Figure 21). A person with known height (Y_{G_T}) was placed in 5 different static positions (a .. e) with precisely measured



(a) Images

(b) Segmentation results

FIGURE 20: Qualitative results for the road segmentation: top row contains the infrared images and the bottom row shows the segmentation results: road pixels are marked with red

depths (Z_{G_T}). For each position of the person, a pair of points, the lowest and the highest image coordinates along the persons vertical median axis, were manually selected and the absolute ($Z_{M_{a_e}}$) and relative ($Z_{M_{r_e}}$) depths errors were estimated (Table 2 - columns 3,4), according to (22) by averaging the results over 3 consecutive static image frames.

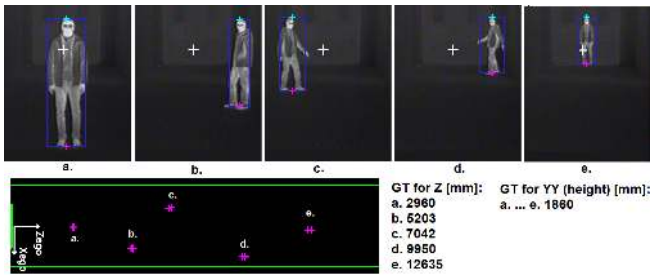


FIGURE 21: Static scenario used for quantitative evaluation of the distance estimator with a pedestrian positioned in 5 known locations

Even a person's height estimation is not relevant for the the pedestrian cross-action recognition problem, its evaluation is an important clue for the overall assessment of the proposed monocular measurement model, since both the depth and height estimations of the objects are very sensitive to the accurate selection of the object's base point (the image projection of the lowest contact point of the object with the ground/road) and its top-most point in the 2D image. Therefore the absolute ($Y_{M_{a_e}}$) and relative ($Y_{M_{r_e}}$) height errors were estimated (Table 3 - columns 3,4) for the same pairs of manually selected points for each person's instance (a .. e), according to (24), again by averaging the results over 3 consecutive static image frames.

The same measurement pattern was performed for lowest and the highest mid points of the 2D bounding box provided

by the pedestrian detector module, and the obtained absolute ($Z_{P_{D_{a_e}}}$) and relative ($Z_{P_{D_{r_e}}}$) depths errors and the obtained absolute ($Y_{P_{D_{a_e}}}$) and relative ($Y_{P_{D_{r_e}}}$) height error are presented in the last 2 columns of Tables 2 and 3, respectively. The errors for the manually selected points (columns 3,4) and for the automatically detected points (columns 5,6) are of the same order, ranging mostly bellow 5% which is more than acceptable for a near-depth range urban scenario, and are comparable with the performances of more precise sensors (i.e. stereo vision [75]), considering the simplified flat road assumption.

#	Z_{G_T}	$Z_{M_{a_e}}$	$Z_{M_{r_e}}[\%]$	$Z_{P_{D_{a_e}}}$	$Z_{P_{D_r}}[\%]$
a	2960	-105	-3.55	-255	-8.60
b	5203	-149	-2.87	31	0.60
c	7042	-115	-1.64	-138	-1.95
d	9950	-70	-0.71	147	1.48
e	12635	-12	-0.09	256	2.03

TABLE 2: Depth evaluation in a controlled scenario

#	Y_{G_T}	$Y_{M_{a_e}}$	$Y_{M_{r_e}}[\%]$	$Y_{P_{D_{a_e}}}$	$Y_{P_{D_r}}[\%]$
a	1860	-33	-1.77	-78	-4.21
b	1860	-54	-2.90	-84	-4.53
c	1860	-49	-2.63	-26	-1.42
d	1860	-12	-0.65	0	-0.02
e	1860	-2	0.11	28	1.49

TABLE 3: Height evaluation in a controlled scenario

For the lateral position estimation (X coordinate), GT data was not acquired for each person's instance from the static (controlled) scenario but the width of the furthest structure visible in the image (the door contour visible in the background of Figure 21) was assessed. So, at approximately 13m depth, for the 2.653m width structure an absolute width error of -38 mm corresponding to a relative width error of -1.43% was obtained. Obviously, accuracy of X coordinates can be offset-ed by the imprecise alignment between the O_CZ_C and O_EZ_E axes (Figure 14) during the sensor's setup, but it can be minimized by carefully align the principal point of the thermal camera (white cross from Figure 21) with the O_EZ_E axis of the ego-vehicle.

The lateral movement of the pedestrian was assessed by computing the relative speed component (between the pedestrian and ego-car) along the O_EX_E axis in a dynamic sequence. The image positions of the mid-bottom point of the tracked pedestrians' bounding-boxes (Figure 22.a), provided by the pedestrian detection and tracking modules, were transformed in metric coordinates (22, 24, 27) and mapped in the top-view image (Figure 22.b). The horizontal component of the pedestrian's speed is computed as the temporal derivative of the X coordinates against the time difference between two consecutive frames ($\Delta t = 40ms \Leftrightarrow fr\_rate = 25fps$):

$$v_{relX} = \frac{X(t) - X(t - \Delta t)}{\Delta t} = \frac{\Delta X \cdot fr\_rate}{1000}[m/s] \quad (29)$$

For the dynamic scenario presented in Figure 22 the horizontal component of the relative speed v_X was computed for

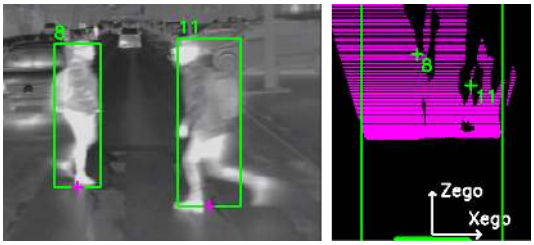


FIGURE 22: Dynamic scenario used for quantitative evaluation of the the horizontal component (along O_EX_E axis) of the relative speed between pedestrians and the ego-car: a. perspective view with tracked bounding-boxes; b. Top view projection of the scene, showing the segmented road surface (magenta) and the reference position of each pedestrian (green cross)

the whole sequence summing about 60 frames. Figures 23 and 24 show the row values (red plots) of the v_X component for two tracked pedestrians crossing the street in front of the ego-car. The row speed components were temporally filtered with a mean low-pass filter of size 5 (green plot) as they are used by the action recognition module. The blue dotted plots represent the average row speed components over the entire sequence and show the general movement behavior of the pedestrians: pedestrian with ID=8 is crossing the street by walking from left to right with an average speed of 1.54 m/s (5.5 km/h) while pedestrian with ID=11 is crossing the street by running from right to left with an average speed of -2.93 m/s (-10.5 km/h).

The difference between the oscillating raw speed and the smoothed one are mainly due to the instantaneous variations of the bottom-center point of the pedestrian (v_1, u_1) as provided by the pedestrian detector, which influence the X coordinate of the pedestrian's position according to (21), (22) and (27) and secondary due to the fact that the pedestrian speed is not constant and follows a stepping induced pattern. However the smoothed speed components have lower standard deviations (0.3 .. 0.4) for both pedestrians being approximately half of the ones computed for the raw speed components (0.6 .. 0.8) and can be used as input features for the cross-action recognition classifier,

D. CROSS ACTION RECOGNITION EVALUATION

For evaluation we have used time series of various lengths extracted from the test dataset. The length of the time series represents the minimum number of frames before the action can be recognized. The metrics used for evaluation are accuracy and F1-score. During the experiments we have varied the length of the time series from 3 to 20 frames and measured the metrics for each length. Table 4 presents these results.

A good accuracy for a time series having a short length means the system can predict the cross or not cross action based on less information about the evolution of the features in time. The accuracy chart and the evolution of the F1-score for various time series lengths is also shown in Figure 25.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

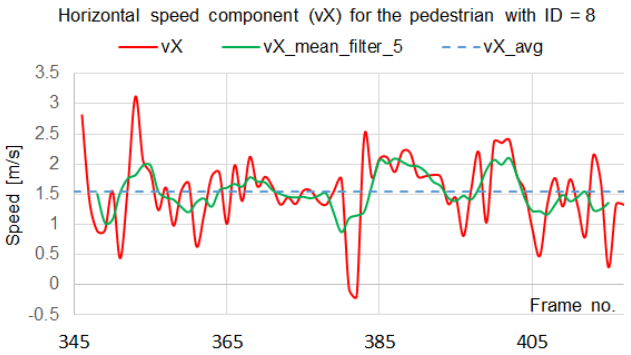


FIGURE 23: Horizontal component of a pedestrian crossing the street from left to right: red plot - raw speed, green plot - filtered speed with a mean low-pass filter of size 5; blue plot - average speed over the entire sequence

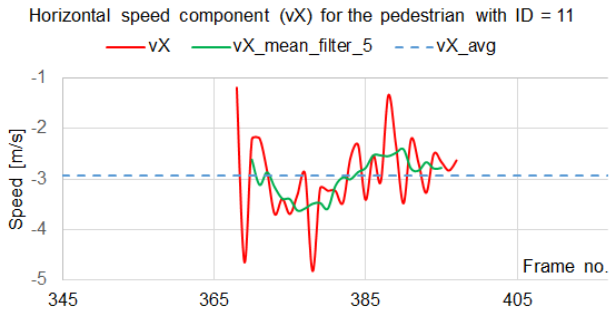


FIGURE 24: Horizontal component of a pedestrian crossing the street from right to left: red plot - raw speed, green plot - filtered speed with a mean low-pass filter of size 5; blue plot - average speed over the entire sequence

Length	Accuracy %	F1-score %	Precision %	Recall %
3	0.9216	0.9311	0.9505	0.9126
4	0.9028	0.9124	0.9572	0.8715
5	0.9038	0.9133	0.9569	0.8735
6	0.9036	0.9129	0.9599	0.8703
7	0.9078	0.9173	0.9551	0.8824
8	0.9122	0.9207	0.9626	0.8824
9	0.9118	0.9200	0.9650	0.8790
10	0.9087	0.9179	0.9535	0.8849
11	0.9156	0.9248	0.9535	0.8978
12	0.9067	0.9143	0.9505	0.8807
13	0.9319	0.9395	0.9528	0.9266
14	0.9105	0.9193	0.9562	0.8851
15	0.9128	0.9212	0.9500	0.8941
16	0.9275	0.9342	0.9595	0.9103
17	0.9184	0.9268	0.9500	0.9048
18	0.9328	0.9412	0.9412	0.9412
19	0.9173	0.9272	0.9333	0.9211
20	0.9085	0.9171	0.9540	0.8830

TABLE 4: Cross action recognition accuracy for various time series lengths

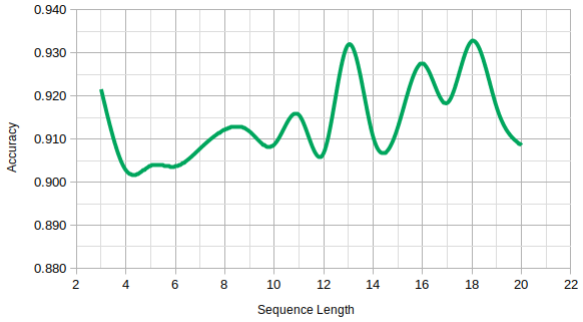


FIGURE 25: Cross action accuracy for various lengths of the time series in the test set

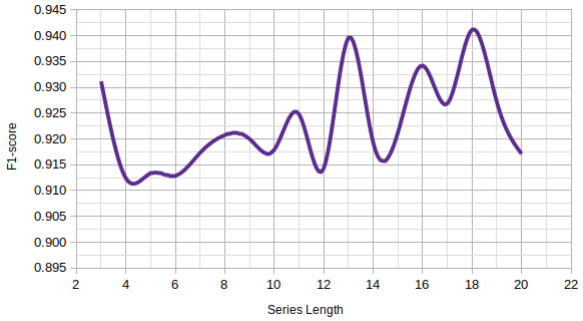


FIGURE 26: Cross action evaluation: F1-score for various lengths of the time series in the test set

It can be noticed that for all time series lengths, varying from 3 to 20, the accuracy is above 90% with a precision higher than 93%. This means the proposed model predicts the pedestrian actions correctly, considering a minimum amount of information gathered for at least 3 frames. The measures presented in Table 4 are average measures for all pedestrian instances.

We also measure the evolution of the cross probability, in relation with pedestrian distance from the car for each of the scenarios depicted in Figures 3 and 4. These measurements are highly dependent on the content of the sequences. Most of the sequences in the CROSSIR dataset contain pedestrians that are at a distance from 3 to 30 meters with respect to the vehicle.

In the scenario with pedestrians walking or running towards the road and crossing continuously not far from the car (2-12 meters) the average cross probability is shown in Figure 27. We also depict the minimum and maximum cross probabilities. It can be noticed from Figure 27 that the minimum cross probability has a value greater than 50%, while the average cross probability for this scenario is close to 90%.

Figure 28 shows the evolution of the cross probability for the scenario in which pedestrians are standing close to the curb and starting to cross. It depicts the average, minimum and maximum cross probabilities. It can be noted that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

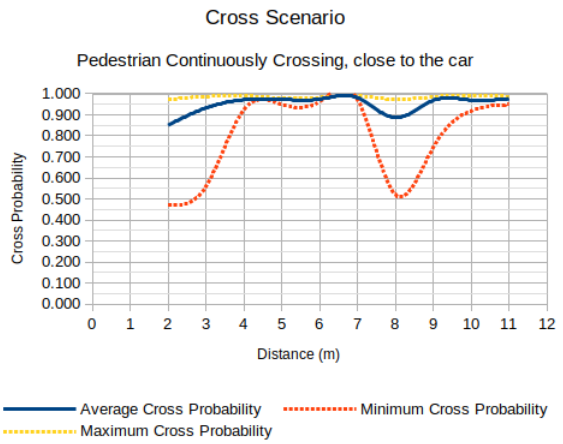


FIGURE 27: Cross action probability for the continuous cross scenario

cross probability starts increasing from a distance of 20 meters. This fact is due to the nature of the test sequences in the dataset for which pedestrians are crossing the street in front of the car, their distance with respect to the car being in the range 5 to 30 meters.

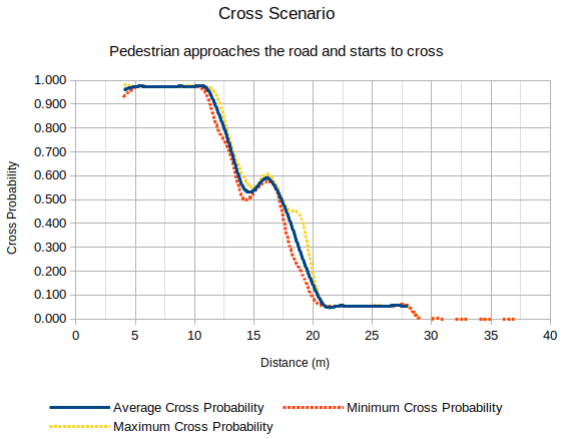


FIGURE 28: Cross action probability for the start to cross scenario

Figure 29 shows the evolution of the cross probability (average, minimum and maximum values) for scenarios in which pedestrians are on the road and start to cross or are already engaged in a cross action. This situation appears when the car turns on a street where a pedestrian is on the road, engaged in a cross action with the direction of movement perpendicular to the direction of the car, or the pedestrian's movement direction is parallel to the direction of the car. It can be noticed that the cross probability in this situation starts to increase if the pedestrian is situated at a distance smaller than 38 meters.

A similar analysis was performed for predicted cross probability on not cross scenarios. When pedestrians are standing,

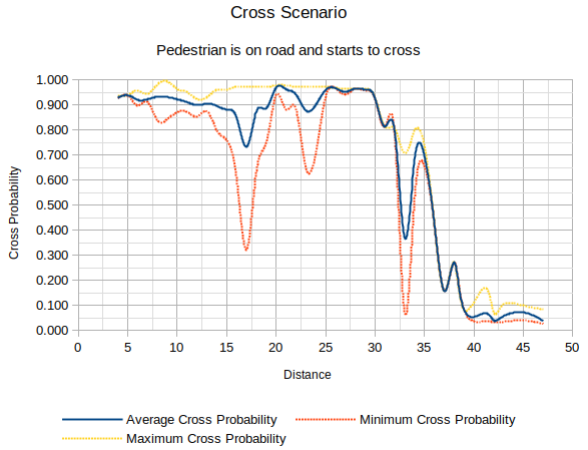


FIGURE 29: Cross action probability for the pedestrian on road cross scenario

walking or running parallel to the road they do not enter the drivable area and their direction of motion is parallel to the road. Figure 30 shows the evolution of the cross probability in this case. It can be noted from Figure 30 that the average

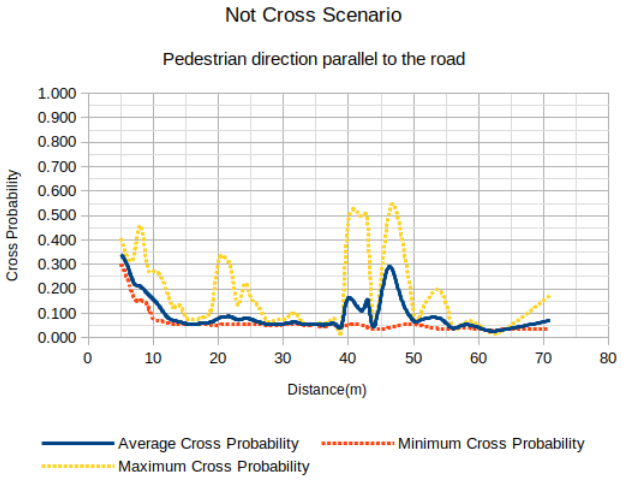


FIGURE 30: Cross action probability for not cross scenario in which the pedestrian is standing next to or moving parallel to the road

cross probability is bellow 40%, which is typical for such situations.

For the case in which pedestrians are walking or running towards the road and then stop without crossing the street the evolution of the cross probability is presented in Figure 31. In this situation it can be noticed that as the pedestrian is getting closer the cross probability increases around of 50% average value and then it decreases. Minimum and maximum cross probabilities are also depicted and it can be observed the maximum cross probability reaching as high as 99% . This case is typical for the sequences in which the pedestrian

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access

comes towards the car quickly and stops in the last moment very close to the car.

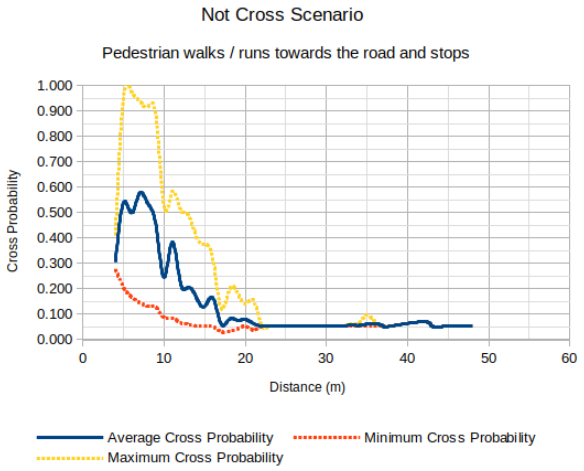


FIGURE 31: Cross action probability for not cross scenario in which the pedestrian is approaching the road and stops

A comparison with other methods is shown in Table 5. Even if those methods are trained for color images, we have used similar algorithms and features that can be computed from the infrared images. For computing the gait information we have employed the pose extraction algorithm proposed by [76], [77], [78]. AlexNet is trained for gait (walking / standing) estimation by [46] and the model is modified in order to provide features on top of which a SVM classifier is trained. We have used as context the road information provided by the semantic segmentation module. Due to the nature of the infrared images we could not include pedestrian crossing signs or traffic lights which are not distinguishable in the heat map of the infrared image.

Method	Accuracy
SVM and gait information [46]	88.75%
LSTM + bounding box information [79]	80.5%
ACF pedestrian detector [6] + LSTM	78.2%
R-CNN pedestrian detector [80] + LSTM	82.2%
Proposed method	93.28%

TABLE 5: Cross action recognition accuracy – comparison with other methods

Two other pedestrian detectors, namely Aggregated Channel Feature (ACF) [6] pedestrian detector and Regions with CNN features (R-CNN) [80] were used as baseline pedestrian detectors with the purpose of result comparison. The pedestrian bounding box information was combined with motion, distance and road features and fed to the LSTM action recognition model.

The execution time for the proposed method was measured on an onboard computer having the following features: i7-3770K CPU with 16GB of memory and an NVidia GeForce RTX 2080 Ti. The execution time for extracting the features provided by the pedestrian detection and tracking module is 26ms, for computing the features given by the road segmentation and pedestrian distance estimation is of 13ms, while the LSTM inference time is of 10 ms.

Figure 32 presents results of the action recognition module in different scenarios. The road segmentation mask is overlapped over the images, the detected and tracked pedestrians are marked with either green (if they do not cross) or red (if they cross), the distance of the pedestrians with respect to the car is shown above the bounding box, while the cross probability is written under the bounding box.

For example the start to cross scenario in which the pedestrian is walking towards the road and he / she starts to cross the road. In Figure 32 a) we show the pedestrian distance with respect to the car and the cross probability. If the action is a not-cross action the pedestrian bounding box is green, and if the pedestrian is crossing the system displays a red bounding box. The continuous cross scenario is shown in Figure 32 b) with pedestrian distances in various ranges: far or closer to the car.

The not cross scenarios when the pedestrian comes towards the road and stops or when the pedestrian is walking parallel to the road are shown in Figure 33 a) and b). Some demonstrative videos and the CROSSIR dataset are available at [2].

VI. CONCLUSION

A modular system for detecting, tracking and recognizing the pedestrians' actions in far infrared images was presented. The contributions of the proposed approach reside in an original time series based cross action recognition model that estimates the pedestrian locations in the scene, their speed and direction of movement, and recognizes with a high accuracy the cross or not cross actions.

The infrared setup is useful for day and night driving conditions, for low visibility environments with fog, snow or heavy rain. The proposed model is based only on the information provided by a monocular infrared camera. Using the known system setup we are able to estimate the pedestrian distance with respect to the ego vehicle. Based on a robust pedestrian detector combined with an original tracking algorithm capable to extract motion and direction information, we integrate road segmentation data, in order to build a time series prediction model that recognizes the pedestrian cross action.

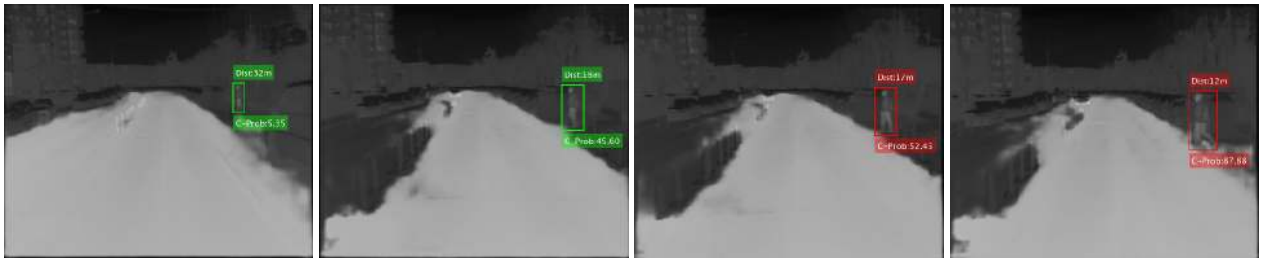
For the evaluation of the model we also propose and share towards the scientific community an annotated dataset, CROSSIR that can be used for pedestrian detection, tracking and action recognition in infrared images. Experiments with various time length series show that the proposed solution achieves an accuracy over 90% for all cross and not cross scenarios captured in the proposed dataset.

REFERENCES

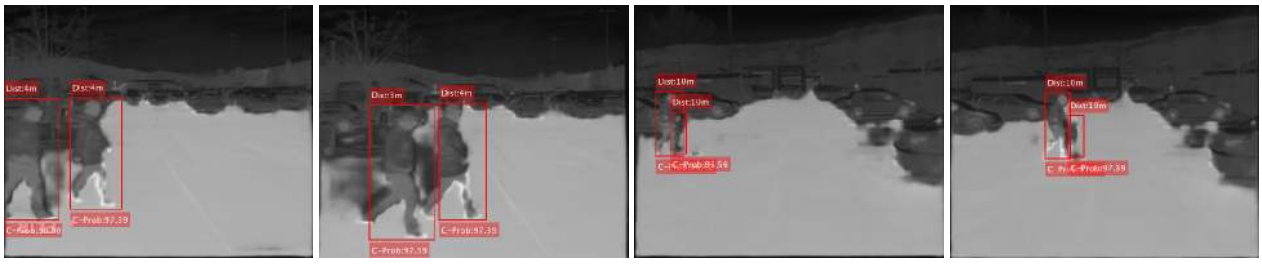
[1] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf, "A literature review on the prediction of pedestrian behavior in urban scenarios," in *2018 21st*

[2] **https://users.utcluj.ro/ raluca/crossir/**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080822, IEEE Access



(a) Start to cross action



(b) Continuous cross

FIGURE 32: Results for cross action recognition: the segmented road is marked with light gray, pedestrians that do not cross are marked with a green bounding box, pedestrians performing a cross action are marked with a red bounding box. The pedestrian distance with respect to the ego-vehicle is noted at the top-left corner of the bounding box, while the cross probability (a number between 0-100) is shown at the bottom left corner of the bounding box.

(a) Not-cross: stop scenario

(b) Not-cross: walk parallel to the road

FIGURE 33: Results for not cross actions: the segmented road is marked with light gray, pedestrians that do not cross are marked with a green bounding box. The pedestrian distance with respect to the ego-vehicle is noted at the top-left corner of the bounding box, while the cross probability (a number between 0-100) is shown at the bottom left corner of the bounding box.

International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3105–3112.

[2] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[3] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 900–918, 2020.

[4] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[5] D. GerÃşnimo, A. M. LÃşpez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[7] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1037–1045.

[8] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 206–212.

[9] R. Brehar, C. Vancea, F. Oniga, M. Negru, and S. Nedevschi, "A study of the impact of hog and lbp based temporal association on far infrared pedestrian detection," in *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2016, pp. 263–268.

[10] Li Zhang, B. Wu, and R. Nevatia, "Pedestrian detection in infrared images based on local shape features," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[11] R. Brehar, C. Vancea, and S. Nedevschi, "Pedestrian detection in infrared images using aggregated channel features," in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2014, pp. 127–132.

[12] R. Brehar and S. Nedevschi, "Pedestrian detection in infrared images using hog, lbp, gradient magnitude and intensity feature channels," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1669–1674.

[13] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[14] C. Li, D. Song, R. Tong, and M. Tang, "Illumination-aware faster r-cnn for robust multispectral pedestrian detection," *Pattern Recognition*, vol. 85, pp. 161 – 171, 2019.

[15] K. N. Renu Chebrolu and P. N. Kumar, "Deep learning based pedestrian detection at all light conditions," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019, pp. 0838–0842.

[16] Z. Cao, H. Yang, J. Zhao, X. Pan, L. Zhang, and Z. Liu, "A new region proposal network for far-infrared pedestrian detection," *IEEE Access*, vol. 7, pp. 135 023–135 030, 2019.

[17] D. Ghose, S. M. Desai, S. Bhattacharya, D. Chakraborty, M. Fiterau, and T. Rahman, "Pedestrian detection in thermal images using saliency maps," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 988–997.

[18] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6517–6525.

[19] R. Brehar, F. Vancea, T. Marita, C. Vancea, and S. Nedevschi, "Object detection in monocular infrared images using classification âĂŞ regresion deep learning architectures," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2019, pp. 207–212.

[20] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *ECCV Workshops*, 2016.

[21] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," *CoRR*, vol. abs/1802.09298, 2018. [Online]. Available: http://arxiv.org/abs/1802.09298

[22] S. Wang and C. C. Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, vol. 122, pp. 484–501, 2016.

[23] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[24] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," *CoRR*, vol. abs/1604.01802, 2016. [Online]. Available: http://arxiv.org/abs/1604.01802

[25] K. Yoon, D. Y. Kim, Y. Young Chul, and M. Jeon, "Data association for multi-object tracking via deep neural networks," *Sensors*, vol. 19, p. 559, 01 2019.

[26] A. Asvadi, P. GirÃčo, P. Peixoto, and U. Nunes, "3d object tracking using rgb and lidar data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1255–1260.

[27] A. Osep, W. Mehner, M. Mathias, and B. Leibe, "Combined image- and world-space tracking in traffic scenes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1988–1995.

[28] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2365–2374.

[29] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3d object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, p. 1110, 02 2020.

[30] P. Li, J. Shi, and S. Shen, "Joint spatial-temporal optimization for stereo 3d object tracking," 2020.

[31] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse, "Pedestrian localization and tracking system with kalman filtering," in *IEEE Intelligent Vehicles Symposium, 2004*, 2004, pp. 584–589.

[32] Xia Liu and K. Fujimura, "Pedestrian detection using stereo night vision," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1657–1665, 2004.

[33] Fengliang Xu, Xia Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 63–71, 2005.

[34] U. Scheunert, H. Cramer, B. Fardi, and G. Wanielik, "Multi sensor based tracking of pedestrians: a survey of suitable movement models," in *IEEE Intelligent Vehicles Symposium, 2004*, 2004, pp. 774–778.

[35] E. Binelli, A. Broggi, A. Fascioli, S. Ghidoni, P. Grisleri, T. Graf, and M.-M. Meinecke, "A modular tracking system for far infrared pedestrian recognition," vol. 2005, 07 2005, pp. 759 – 764.

[36] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, W. Liu, and Y. Liang, "Multi-task driven feature models for thermal infrared tracking," in *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[37] J. Kwak, B. C. Ko, and J. Y. Nam, "Pedestrian tracking using online boosted random ferns learning in far-infrared imagery for safe driving at night," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 69–81, 2017.

[38] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf, "A literature review on the prediction of pedestrian behavior in urban scenarios," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3105–3112.

[39] S. Neogi, M. Hoy, W. Chaoqun, and J. Dauwels, "Context based pedestrian intention prediction using factored latent dynamic conditional random fields," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–8.

[40] Z. Fang and A. M. LÃşpez, "Intention recognition of pedestrians and cyclists by 2d pose estimation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[41] F. Schneemann and P. Heinemann, "Context-based detection of pedestrian crossing intention for autonomous driving in urban environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2243–2248.

[42] J.-Y. Kwak, B. C. Ko, and J.-Y. Nam, "Pedestrian intention prediction based on dynamic fuzzy automata for vehicle driving at nighttime," *Infrared Physics & Technology*, vol. 81, pp. 41 – 51, 2017.

[43] P. Xue, J. Liu, S. Chen, Z. Zhou, Y. Huo, and N. Zheng, "Crossing-road pedestrian trajectory prediction via encoder-decoder lstm," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2027–2033.

[44] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo, "Pedestrian intention and pose prediction through dynamical models and behaviour

classification," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 83–88.

[45] K. Saleh, M. Hossny, and S. Nahavandi, "Intent prediction of vulnerable road users from motion trajectories using stacked lstm network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 327–332.

[46] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[47] Z. Xu, J. Zhuang, Q. Liu, J. Zhou, and S. Peng, "Benchmarking a large-scale fir dataset for on-road pedestrian detection," *Infrared Physics & Technology*, vol. 96, pp. 199–208, 2019.

[48] FLIR, "Flir thermal datasets for algorithm training."

[49] Q. Liu, Z. He, X. Li, and Y. Zheng, "Ptb-tir: A thermal infrared pedestrian tracking benchmark," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 666–675, 2020.

[50] Intel. (2019) Computer vision annotation tool: A universal approach to data annotation. [Online]. Available: https://software.intel.com/content/www/us/en/develop/articles/computer-vision-annotation-tool-a-universal-approach-to-data-annotation.html

[51] O. Lahdenoja, J. Poikonen, and M. Laiho, "Towards understanding the formation of uniform local binary patterns," *ISRN Machine Vision*, vol. 2013, 07 2013.

[52] G. FarnebÃd'ck, "Two-frame motion estimation based on polynomial expansion," vol. 2749, 06 2003, pp. 363–370.

[53] H. W. Kuhn and B. Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart*, pp. 83–97, 1955.

[54] E. Romera, J. M. ÃAlvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1789–1794.

[55] ——, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, Jan 2018.

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with lstm," in *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, vol. 2, 1999, pp. 850–855 vol.2.

[58] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[59] MATLAB, *version 9.7.0 (R2019b)*. Natick, Massachusetts: The MathWorks Inc., 2019.

[60] S. Nedevschi, R. Danescu, F. Oniga, and T. Marita, *Tehnici de viziune artificiala aplicate in conducerea automata a autovehiculelor*. Cluj-Napoca: U.T. Press, 2012.

[61] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," 2017.

[62] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.490

[63] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," 2018.

[64] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr, "Staple: Complementary learners for real-time tracking," 2016.

[65] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, and H. Wang, "Learning deep multi-level similarity for thermal infrared object tracking," 2019.

[66] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," 2017.

[67] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," 2018.

[68] Y. Wang, H. Huang, X. Huang, and Y. Tian, "Eco-hc based tracking for ground moving target using single uav," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 6414–6419.

[69] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," 2019.

[70] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen, "Deep convolutional neural networks for thermal infrared object tracking," *Knowledge-Based Systems*, vol. 134, pp. 189 – 198, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705117303544

[71] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. Yang, "Hedged deep tracking," in *Proceedings - 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, ser. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. United States: IEEE Computer Society, Dec. 2016, pp. 4303–4311, publisher Copyright: © 2016 IEEE. Copyright: Copyright 2017 Elsevier B.V., All rights reserved.; 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016 ; Conference date: 26-06-2016 Through 01-07-2016.

[72] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical convolutional features for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3074–3082.

[73] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," 2016.

[74] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[75] T. Marita, F. Oniga, S. Nedevschi, and T. Graf, "Calibration accuracy assessment methods for stereovision sensors used in vehicles," *in Proceedings of IEEE 3-rd International Conference on Intelligent Computer Communication and Processing*, no. 1, pp. 111–118, 2007.

[76] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.

[77] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," *arXiv preprint arXiv:1812.00324*, 2018.

[78] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient online pose tracking," in *BMVC*, 2018.

[79] D. O. Pop, A. Rogozan, C. Chatelain, F. Nashashibi, and A. Bensrhair, "Multi-task deep learning for pedestrian detection, action recognition and time to cross prediction," *IEEE Access*, vol. 7, pp. 149 318–149 327, 2019.

[80] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

RALUCA DIDONA BREHAR received the B.S., M.S. and PhD degrees in computer science from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 2006, 2008 and 2015, respectively. She is currently an Associate Professor in Computer Science with the Department of Computer Science, TUCN. Her main research interests and directions are on pedestrian detection in color and infrared images, medical image processing of ultrasound images and deep learning in computer vision. She is a member of the Image Processing and Pattern Recognition Research Center from TUCN and was involved in numerous international research projects (Horizon 2020, FP7, with third parties-Volkswagen AG) national founded ones in the field of vision based driving assistance systems and medical imaging.

MIRCEA PAUL MURESAN received his B.Sc. in Computer Science in 2014 and his M.Sc. in Artificial Intelligence and Computer Vision in 2016 from the Technical University of Cluj-Napoca (TUCN). He is currently pursuing a PhD degree in Computer Science at TUCN. Mircea Paul Muresan is also an Assistant Professor in Computer Science with the Department of Computer Science,TUCN, from 2019. He is a member of the Image Processing and Pattern Recognition Research Center from TUCN, and has been involved in multiple national and international research projects. His main research interests include stereo vision, data association and tracking, sensor fusion, autonomous systems, embedded computer vision and industrial inspection.

TIBERIU MARIŢA has an experience of over 20 years of research in the field of computer vision, targeting the following topics: camera calibration, real time applications for driving assistance and medical imaging. The research activity was materialized by publishing over 70 publications in the field. He has been involved as an active member in several international research projects (Horizon 2020, FP7, with third parties-Volkswagen AG) and national founded ones in the field of vision based driving assistance systems and medical imaging.

CRISTIAN-COSMIN VANCEA received the PhD degree in computer science from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, in 2020. He is currently a Senior Lecturer in Computer Science with the Department of Computer Science, TUCN. His main research is focused on camera calibration, image processing and environment perception for automated driving with deep learning and computer vision. He is a member of the Image Processing and Pattern Recognition Research Center from TUCN with current and past activity in multiple national and international research projects.

MIHAI NEGRU received the B.S., M.S. and PhD degrees in computer science from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 2006, 2008 and 2015, respectively. He is currently an Associate Professor in Computer Science with the Department of Computer Science, TUCN. His main research focus is on environment perception for advanced driving assistance systems in intelligent vehicles and automated driving. He is a Member of the Image Processing and Pattern Recognition Research Center from TUCN. He was involved in more than 20 national and international research projects, 3 as coordinator. His research interests include computer architecture, digital signal processing, image processing, computer vision and embedded computer vision.

SERGIU NEDEVSCHI (M99) received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Cluj-Napoca (TUCN), Romania, in 1975 and 1993 respectively. From 1976 to 1983 he was a Researcher with the Research Institute for Computer Technologies Cluj-Napoca and since 1983 joined the TUCN. In 1998 he was appointed as a full Professor of Computer Science, he founded and since then he is leading the Image Processing and Pattern Recognition Research Center. From 2000 to 2004 he was the Head of the Computer Science Department, from 2004 to 2012 he was the Dean of the Faculty of Automation and Computer Science and from 2012 to 2020 the Vice-Rector with Scientific Research of TUCN. His research interests include image processing, pattern recognition, computer vision, machine learning, intelligent and autonomous vehicles. He was involved in more than 80 research projects, being the coordinator of 62 of them. The industrial cooperation with important automotive players like as Volkswagen AG, Robert Bosch GmbH, SICK AG and research institutes such as VTT, INRIA was achieved through funded research projects. He has published more than 400 scientific papers and has edited over 20 volumes, including books and conference proceedings.