Multi-Task Deep Learning for Pedestrian Detection, Action Recognition and Time to Cross Prediction

Danut Ovidiu Pop, Alexandrina Rogozan, Clément Chatelain, Fawzi Nashashibi, Abdelaziz Bensrhair

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 30, 2019, accepted September 11, 2019, date of publication October 1, 2019, date of current version October 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944792

Multi-Task Deep Learning for Pedestrian Detection, Action Recognition and Time to Cross Prediction

DĂNUȚ OVIDIU POP[1,2,3], ALEXANDRINA ROGOZAN[2], CLEMENT CHATELAIN[2], FAWZI NASHASHIBI[1], AND ABDELAZIZ BENSRHAIR[2]

[1]INRIA Paris - RITS Team, 75012 Paris, France
[2]Normandie Université - INSA Rouen, LITIS, 76800 Rouen, France
[3]Department of Computer Science, Babeş University, 400084 Cluj-Napoca, Romania

Corresponding author: Dănuţ Ovidiu Pop (danut-ovidiu.pop@inria.fr)

This work was supported by the Human Inspired Autonomous Navigation in Crowds (HIANIC) Project.

ABSTRACT A pedestrian detection system is a crucial component of advanced driver assistance systems since it contributes to road flow safety. The safety of traffic participants could be significantly improved if these systems could also predict and recognize pedestrian's actions, or even estimate the time, for each pedestrian, to cross the street. In this paper, we focus not only on pedestrian detection and pedestrian action recognition but also on estimating if the pedestrian's action presents a risky situation according to time to cross the street. We propose 1) a pedestrian detection and action recognition component based, on RetinaNet; 2) an estimation of the time to cross the street for multiple pedestrians using a recurrent neural network. For each pedestrian, the recurrent network estimates the pedestrian's action intention in order to predict the time to cross the street. We based our experiments on the JAAD dataset, and show that integrating multiple pedestrian action tags for the detection part when merge with a recurrent neural network (LSTM) allows a significant performance improvement.

INDEX TERMS Action recognition, deep learning, pedestrian detection, time-to-cross estimation.

I. INTRODUCTION

Pedestrian detection is one of the highly debated issues in the intelligent systems field due to its large-scale applicability in self-transportation and driver assistance systems. It is one of the main interests of transportation safety as it could lead to a significant reduction in the number of traffic accidents and ensure the safety of pedestrians, who are the most vulnerable road users.

Human errors abound due to fatigue, driving the car while using the telephone, driving under the influence of medicine, or pedestrians' bad and/or risky behavior, any of which may generates traffic collisions. These collisions between cars and pedestrians could be greatly decreased if human error could be eliminated by employing an Advanced Driver Assistance System (ADAS) for pedestrian detection.

Automotive companies like BMW, Mercedes, Nissan, Audi, Toyota, have this ADAS technology in the majority of their high-end automobiles.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengbing He.

This issue has been widely investigated, but it still remains an open challenge because progress in pedestrian detection is hindered by the difficulty of detecting all (partially) occluded pedestrians and the problem of operating efficiently in severe weather conditions. Moreover, current systems cannot yet understand the intention of road users involved to ensure their safety and secure the traffic flow. For this purpose, the system should have i) a detection model for localizing and recognizing the pedestrians among other road users, ii) a classification model to distinguish the pedestrian actions, and iii) a prediction model to estimate the pedestrian actions over the next frames (short, medium and/or long-time prediction). The prediction component should perform efficiently in various environmental circumstances and even offer the possibility of estimating the time to cross the street for each pedestrian.

The difficulty in solving these problem comes from the lack of public annotated data bases. Hence, there are no public databases annotated with pedestrian time to cross, while there are several interesting huge pedestrian detection databases (Kitti, Caltech, among others). Another problem is that those databases do not provide any pedestrian

action labels. To the best of our knowledge, the only public data set with pedestrian action tags in urban traffic environmental is JAAD [1]. Because this data set does not provide the annotations directly for pedestrian time to cross, we determine it for each pedestrian trajectory (enable the correspond frames label sequences).

The question is, could we manage the pedestrian action classification and the pedestrian bounding box (BB) detection in one end-to-end detector? or we must use two separate models: first for pedestrian detection and then for pedestrian action recognition, as existing approaches from literature?

The contribution of this paper concerns solving this issue by applying a multi-task deep learning model for detecting, classifying, and estimating the time to cross for multiple pedestrian actors.

To do so, we develop the following methodology relying on a deep learning approach:

- Train all pedestrian Bounding Boxes (BB) samples with the RetinaNet [2] for pedestrian detection purposes;
- Split the pedestrian Joint Attention for Autonomous Driving (JAAD) [1] data set into four classes for pedestrian action functionality: pedestrian is preparing to cross the street, pedestrian is crossing the street, pedestrian is about to cross the street, and pedestrian intention is ambiguous;
- Train a Long Short-Term Memory (LSTM) model using only BB coordinates in order to estimate the time to cross of each pedestrian.

The paper is organized as follows: Section 2 outlines some existing approaches from the literature and gives our main contribution. Section 3 presents an overview of our system. Section 4 describes the experiments setups. Section 5 shows the results on the JAAD dataset. Finally, Section 6 presents our conclusions.

II. RELATED WORK

Several research activities addressing pedestrian detection have produced significant performances for this issue [3]–[7]. The estimation of the pedestrian intention, and especially of the pedestrian risky actions is even more challenging because of the ambiguities in pedestrian motions. Indeed, the pedestrian could decide to change its behavior/movement in less than one second, which increases the difficulty of solving the problem. Nevertheless, the interest in estimating the pedestrian actions for smart cars has significantly increased in the last years [8]–[12]. In order to find a solution to this issue, the research analyzed various features like pedestrian movements and/or pedestrian behaviors [13], [14], interactions between pedestrians [15], [16] and pedestrian tracking paths [9], [10].

A comprehensive review of the predicting pedestrian behavior research is presented in [12], which includes several pedestrian action and movement estimation approaches, and also sets out the advantages and shortcomings of the currently available datasets. The authors assume that the prediction of pedestrian intention requires to use pedestrian specific dynamic information as well as contextual road environment. In [17], the authors present a pedestrian action recognition approach based on AlexNet handling JAAD dataset, where they investigate whether the full pedestrian body and part of the pedestrian body (consisting either of the head or lower pedestrian body) influence the classification task. They also use a linear SVM to distinguish the situation of a pedestrian crossing or not the street based on pedestrian attention information. The authors also conclude that it is better to use temporal and spatial-temporal contextual information to increase the prediction performance.

A pedestrian position estimation based on the Extended Kalman Filter (EKF) and Interacting Multiple Model (IMM) algorithm using Constant Velocity, Constant Acceleration and Constant Turn is proposed in [9]. The authors also build a dataset, the Daimler data set, with four pedestrian actions called: crossing, bending in, bending out, and stopping. A combination of the Gaussian Process Dynamical Models, Probabilistic Hierarchical Trajectory Machine with Kalman Filter and Interacting Multiple Model-based on the Daimler data set using stereo vision images is presented in [18]. The authors get better performance than the EKF-IMM model for the stopping situations. They also make a comparison between these approaches and conclude that the performances are almost similar for other pedestrian actions.

A short-term prediction of pedestrian behaviors using Daimler datasets was included in [8]. It is based on a Variational Recurrent Neural Network which provides the latent variables suitable for a dynamic state-space model. The authors predict whether a pedestrian is about to stop or to cross, and obtain high performance on the Daimler benchmark. To predict the pedestrian trajectory and its final destination, an approach using CNN base on LSTM and path planning is presented in [10]. This system can predict destinations and pedestrian trajectories. A mixture of CNN-based pedestrian detection, tracking and pose estimation to predict the pedestrian crossing actions based on the JAAD dataset is addressed in [13]. The authors utilize the Faster R-CNN object detector based on VGG16 CNN architecture for the classification task, use a multi-object tracking algorithm based on the Kalman filter, apply the pose estimation pattern on the bounding box predicted by the tracking system, and finally use an SVM or a Random Forest to classify the pedestrian actions (Crossing /Not Crossing).

All these approaches for pedestrian action prediction exploit a standard pedestrian detection component which only discriminates between the pedestrian from non-pedestrian, among other road users, and estimates the pedestrian action or its final destination for the next frames (short, medium and long term) for all detected pedestrians. The time to cross estimation of pedestrians is more challenging than predicting the pedestrian action since it requires contextual spatial-temporary: a fine analysis of the whole scene, as well as a fine analysis of the pedestrian motion. Let us emphasize that this task is challenging even for human beings.

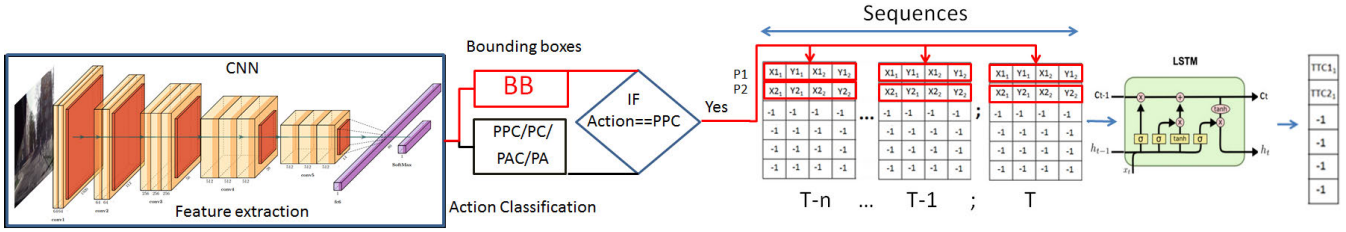


FIGURE 1. Our time to cross the street estimation method using only BB coordinates in order to estimate the time to cross the street. BB = Bounding Box coordinates, PPC = Pedestrian is Preparing to Cross the street; PC = Pedestrian is crossing the street; PAC = Pedestrian is About to Cross the street; PA = Pedestrian's intention is Ambiguous; P1, P2 = Detected Pedestrians; TTC=Time to cross; −1 = no pedestrian.

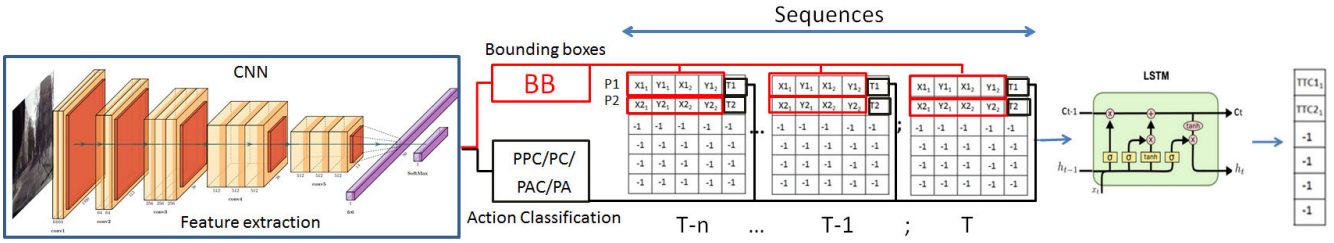


FIGURE 2. Our time to cross the street estimation method using the BB coordinates and pedestrian action labels in order to estimate the time to cross the street. BB = Bounding Box coordinates, PPC = Pedestrian is Preparing to Cross the street; PC = Pedestrian is crossing the street; PAC=Pedestrian is About to Cross the street; PA = Pedestrian's intention is Ambiguous; P1, P2 = Detected Pedestrians; T1, T2 = Pedestrian Action Tags; TTC = Time to cross; −1 = no pedestrian.

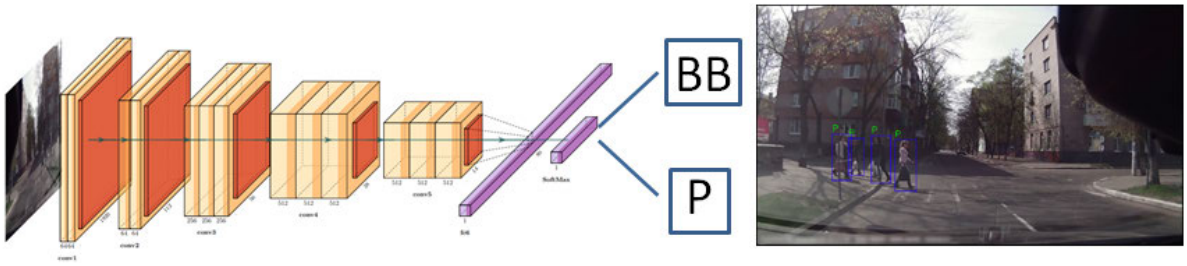


FIGURE 3. The classical pedestrian detection method. BB = Bounding Box coordinates; P = Pedestrian.

To our knowledge, there are no different approaches for pedestrian time to cross (TTC) prediction, other than the method addressed in [13] on JAAD dataset. Nevertheless, the authors in [13] have handled this problem in a step-by-step manner, including the pedestrian tracking component, based on different image processing and machine learning approaches, allowing finally for the pedestrian short-term (1 frame) TTC prediction. We propose an original method for TTC prediction, without an explicit tracking component, based only on deep learning neural networks, allowing a short, medium and long term prediction. Moreover, there is a lack of public annotated data. The JAAD dataset is not annotated for the prediction of pedestrian time to cross issue. The issue of TTC prediction is addressed in [13] where the authors made their own pedestrian TTC annotation on JAAD dataset to solve it, but the authors did not make public these annotations. Further, the authors did not apply their annotation process on all JAAD videos, but only on several sequences. For the TTC prediction problem, we select some cues from the JAAD [1] public data set in order to solve this issue and then we made our pedestrian TTC annotation for all videos.

We also present a multi-task application which can estimate the time to cross the street for each pedestrian using a recurrent neural network (LSTM) in two ways:

- using only BB coordinates in order to estimate the time to cross the street (see Figure 1);
- using BB coordinates and pedestrian action tags in order to estimate the time to cross the street (see Figure 2);

We use the classical approach where the detection and prediction part were independently analyzed (we called the two-stage approach). The LSTM estimate the time to cross the street for each video sequence (estimate the time to cross for all pedestrians from the entire visual spectrum).

Since the detection and estimation components are connected, we also investigate the pedestrian detection issue in two ways:

- pedestrian detection using only the pedestrian BB, which we call the classical method (see Figure 3);
- pedestrian detection and action recognition using the pedestrian BB with the corresponding pedestrian action labels: Pedestrian is Preparing to Cross the street (PPC), Pedestrian is crossing the street (PC), Pedestrian is about

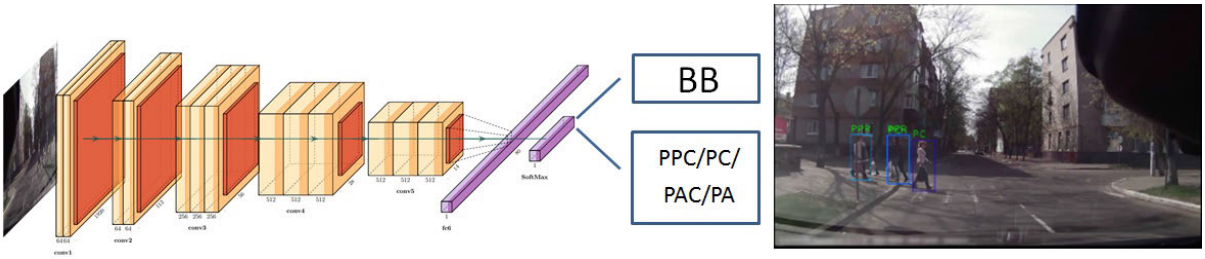


FIGURE 4. The classical pedestrian detection method. BB = Bounding Box coordinates; PPC = Pedestrian is Preparing to Cross the street; PC = Pedestrian is crossing the street; PAC = pedestrian is about to cross the street; PA = pedestrian intention is ambiguous.

to cross the street (PAC), and pedestrian intention is ambiguous (PA) (see Figure 4).

III. METHODOLOGY

In this section, we outline the components and methods used to solve these issues.

A. PEDESTRIAN DETECTION

In order to develop a pedestrian detection system, it is mandatory to take into account three main components: the sensors employed to capture the visual road environment, the image processing elements, and the classification parts. In general, all these components are correlated to achieve a high detection performance, but specific items are seldom investigated independently according to the target application. We have concurrently examined the detection part by applying a generic object detector based on the public RetineNet [2]. We have handled the Resnet50 [19] CNN architecture for the classification task with the Keras public open-source implementation described in [2]. All the training process is based on the JAAD [17] dataset, which provides an annotation of pedestrians with behavioral tags and pedestrians without behavior tags.

The Jaad data set [17], [20] descriptions and annotations present various specific events and actions made by pedestrians before crossing the street, that allow us to divide the pedestrian actions into four classes, according to our goal, the TTC estimation: pedestrian is preparing to cross the street (PPC), pedestrian is crossing the street (PC), pedestrian is about to cross the street (PAC), and pedestrian intention is ambiguous (PA).

We adopted two approaches in the training stage:

1) using all pedestrian BB samples where we consider all the pedestrian annotation tags as a pedestrian (P);
2) using the four proposed pedestrian action tags mentioned above and taking into account only the pedestrian behaviors (PPC, PC, PAC, PA).

B. ESTIMATION OF PEDESTRIAN TIME TO CROSS

The estimation of time to cross for each pedestrian is essential for the ADAS systems since it could predict if and when there could be a risky situation.

From a machine learning point of view, TTC estimation can be considered as a regression problem, where we aim at estimating the remaining time or real value (whether we consider several frames or time in seconds) for each frame of a video. As the dynamic of the signal is essential to estimate TTC efficiently, we have naturally turned toward the use of a recurrent neural network to capture the temporal context of the motion. Among recurrent models, we have chosen to use LSTMs which have shown their efficiency on many sequence analysis problems. For instance, it was shown in [21] that RNNs improve signal estimation compared to the Kalman filter.

The TTC estimation of a video can be achieved regarding two strategies:

- an individual estimate for each pedestrian BB sequence provided by the pedestrian detector (using only PPC samples);
- multiple estimates for all detected pedestrians (using all samples).

We emphasis that the detection and prediction components are trained independently.

The detection step is based on RetinaNet [2], because its performance exceeds the Faster R-CNN [22], R-FCN [23], SSD [24] and YOLOv1 [25]. It has as input the entire RGB images and returns for each pedestrian the corresponding bounding box and its action tag.

The prediction model is based on LSTM, and it has the 2D bounding box (BB) coordinates as input data provided by the detection component. The output consists of time to cross for each pedestrian, and it outlines over how many frames the pedestrian crosses the road. We take into account the temporal context information for the previous frames from T-5, T-14, and T-40 in order the estimate the time to cross the street in term of short (5 frames), medium (14 frames) and long (40 frames) term estimation. Our prediction model consists of four blocks of LSTM with 50 nodes each followed by dropout layer with 20% drop off rate for each LSTM layer, and finally, two fully connected layers with 20 neurons. We used this architecture because we observed a better performance on these settings. That have been tuned over a validation dataset.

IV. EXPERIMENTS

In this section, we present our set of experiments, including setups and performance assessment of our approaches.

A. DATA SETUP

The experiments were performed on the JAAD dataset [17] since its data was collected in usual urban road traffic environments for different locations, times of the day, road and weather conditions. This dataset provides pedestrian bounding boxes (BB) for pedestrian detection (including, for several of them, the pedestrian action tags), pedestrian attributes for estimating the pedestrian behavior and traffic scene elements. It has 346 video sequences (between 5 and 15 seconds long) with an image resolution of 1920 × 1080 pixels and respectively 1280 × 720 pixels recorded in different urban environments [1]. Moreover, it contains approximately 337k pedestrian BB samples, of which around 72.000 (18%) samples are tagged as partially occluded BBs and 46000 (11%) samples as heavily occluded BBs. We use all the pedestrian BB samples, including the partially and heavily occluded pedestrians for all training and testing process.

B. TRAINING PROTOCOL

We used the first 250 video sequences of JAAD data set for the training process and the rest for the testing. We used 10% from training set for validation process.

The training and testing samples include even the partially occluded and heavily occluded BBs.

In [17], [20], the authors present a variety of pedestrian behaviors done before crossing and after crossing the street and even when the pedestrian does not cross the street. These behaviors were collected and annotated with different action labels according to the pedestrian events for each pedestrian from all video sequences.

The events could be:

- the pedestrian completes to cross the street;
- the pedestrian has no intention to cross the road (e.g. sits on a public bench, waiting for public transportation);
- the pedestrian does not cross the street (e.g the pedestrian has started to cross the street, but suddenly he/she is stopping).

For instance, if the pedestrian is going to cross the street, the pedestrian is doing at least one action, like standing, looking, and then crossing the street, or moving, looking, and then crossing the street. The pedestrian actions done before or during one event, could be different for each pedestrian, even if the event is the same. Hence, according to these action annotations, we can observe that there exists a specific pattern for each pedestrian for a given event.

Therefore, according to the specifications and annotations presented above, we separate the pedestrian action labels into four classes:

1) Pedestrian is Preparing to Cross the street (PPC), where the pedestrian is walking/standing, whatever the pedestrian pays attention or not, and changes or does not change behavior before crossing. In this case, the pedestrian is definitively assumed to cross the street after these actions. The problem is to estimate precisely the time to cross for preventing a collision.
2) Pedestrian is Crossing the street (PC), where the pedestrian is observed from the point of crossing until the pedestrian has crossed the road. There are video sequences beginning directly from the point of crossing the street.
3) Pedestrian is About to Cross the street (PAC), where the pedestrian is about to cross and pays attention and acts according to the event. Therefore, the pedestrian will not always cross the street after this action.
4) Pedestrian intention is Ambiguous (PA), where the pedestrian is walking/standing, and his/her intention is ambiguous. In this case, the pedestrian has crossed the road or perform another event which does not present a risk situation.

1) THE DETECTION LEARNING PROTOCOL

We train the Convolution Neural Network (CNN) in a few ways:

1) We train the CNN with all pedestrian samples where we consider all the action tags as a pedestrian (P);
2) We train the CNN with the pedestrian action tags mentioned above (PPC, PC, PAC, PA);

The first 250 videos (with the original resolution of 1920 × 1080 pixels) are used for the training process. We have 95170 pedestrian samples with actions tags of which 24324 are preparing to cross the street (PPC), 51012 pedestrian samples are crossing the street (PC), 14267 pedestrian samples are about to cross the street (PAC), and 5567 pedestrian samples whose intentions are ambiguous (PA).

We perform the CNN learning process during 48 hours on 2 GPU, with a batch size of 1, using an initial learning rate value of 0.0005 with ADAM algorithm learning.

2) THE TIME-TO-CROSS ESTIMATION LEARNING PROTOCOL

The pedestrian time to cross was calculated only for pedestrians who are preparing to cross the street (PPC) because only in this case are the pedestrians definitively going to cross the street and only in this particular case can we estimate the time to cross for each pedestrian. Thus after a PA action, the pedestrian will never cross the street, and after a PAC, the crossing is quite unpredictable (even for the pedestrian itself), but they can start directly with PC.

To determine the time to cross, we use an LSTM which is trained independently of the CNN based detector since it is applied after the detection step.

We create a bounding box matrix to predict the time to cross for multiple pedestrians sequences within the same LSTM (see Figure 5). The LSTM was trained using the following methodology:

- We created an input bounding box matrix (4 × 20) for each frame T where we set the bounding box coordinates only for PPC pedestrian samples or set for all pedestrian samples. For the PPC element in the input matrix, the corresponding output is the time to cross, which consists in the descending scrambling order of

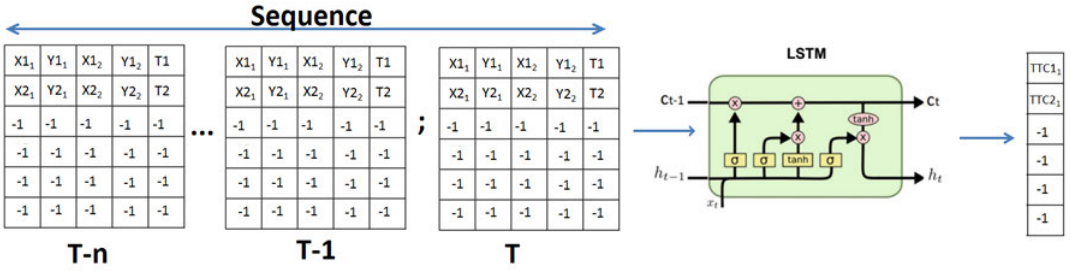


FIGURE 5. The proposed LSTM based architecture for pedestrian time to cross estimation. Input: The BB matrix (4 × 20) at frame T until previews T-n (n = 5, 14, 40), where the X_{i1}, Y_{i1}, X_{i2}, Y_{i2} i = 1 to 20 are the BB coordinates for each pedestrian i detected on frame T; output: TTC(i) = number of frames from frame T to the beginning of crossing for the pedestrian i; −1 = no pedestrian.

TABLE 1. The detection-classification performances. The labels represent: P = Pedestrian, PPC = Pedestrian is Preparing to Cross the street, PC = Pedestrian is Crossing the street, PAC = Pedestrian is About to Cross the street, and PA = Pedestrian intention is Ambiguous.

Learning On	P	PC	PPC	PAC	PA	mAP
	AP	AP	AP	AP	AP	±CI
All pedestrian samples	56.05% ±0.93	x	x	x	x	56.05% ±0.93
Pedestrian with Action Tags	x	65.57% ±1.35	17.67% ±1.36	13% ±1.54	9.22% ±1.63	26.36% ±0.83

frames to the moment of crossing. While for the other pedestrians (PA, PAC, PC), the corresponding output is (−1). In our approach, we consider there are no more than 20 pedestrians per frame (see Figure 1).

- We created an input bounding box matrix (5 × 20) for each frame where we set the bounding box coordinates and pedestrian action tag only for PPC pedestrian samples or set for all pedestrian samples. The input matrix values for the pedestrian action tags are coded with the following: PPC = 0; PA = 1; PC = 2; PAC = 3. The output matrix is the time to cross for the PPC tag, while for the other pedestrians (PA, PAC, PC) the corresponding output is (−1) (see Figure 2).

We performed the LSTM training process with the ADAM learning algorithm method, using previous time steps of 5, 14, and respectively 40 frames to estimate time to cross. For each step, the LSTM estimates over how many frames the PPC pedestrian will cross the street.

C. TESTING PROTOCOL

The testing set used to assess the CNN based detector, and the LSTM based predictor performances are independent of the training dataset. It contains 105 video sequences. It has a total of 43420 pedestrian samples of which 12110 samples are pedestrians who are preparing to cross the street (PPC), 19157 samples are pedestrians who are crossing the street (PC), 1296 samples are pedestrians who are about to cross the street (PAC) and 4857 examples where their intention is ambiguous (PA).

We test the prediction component in two different ways:

- first only on the 12110 pedestrian samples to assess only the predictor capabilities independently of the pedestrian detector and classifier, because this is the only case where the pedestrians are clearly going to cross the street;
- second on all 43420 pedestrian samples in order to assess the overall performance of the system.

Our testing methodology generally consists of the upcoming plan:

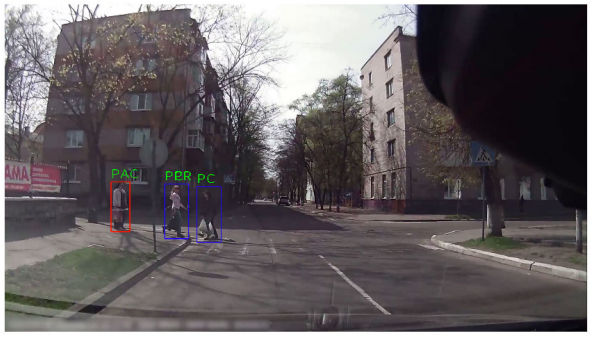
- testing only the pedestrian detection;
- testing the pedestrian detection with action classification capability;
- testing, independently of the detection and classification components, the pedestrian time to cross estimation on the PPC pedestrian samples only;
- testing, independently of the detection and classification components, the pedestrian time to cross estimation on all real pedestrian samples;
- testing the detection component connected with the prediction component (time to cross).

In this paper, we performed all these testing steps.

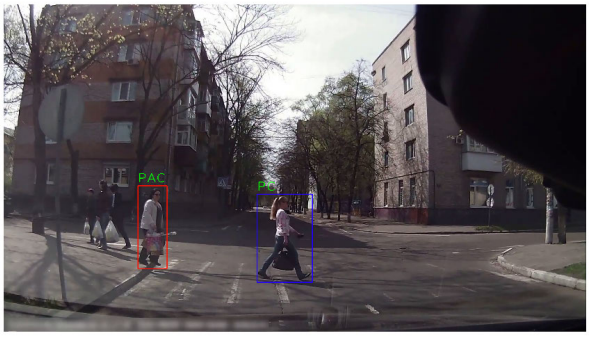
D. EVALUATION PROTOCOL

The evaluation process for all the CNN based detector-classifier models was done with the Tensorflow Deep Neural Network Framework. The performances were assessed by the average precision (AP) and mean average precision (mAP) for the detection part. The AP and mAP values were computed using the TensorFlow metrics tool. The AP is calculated for each class, where a detection is considered accurate if the BB detection-result is higher than 50% (Intersection over Union, IoU $\geq$ 0.50).

Moreover, we use the Root Mean Square Error (RMSE) using the Scikit-Learn tool [26], in order to measure the

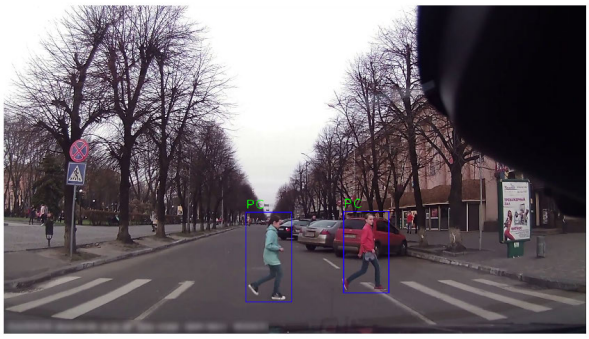


(a) Results of pedestrian detection using PPC, PC, PAC, PA action classification.



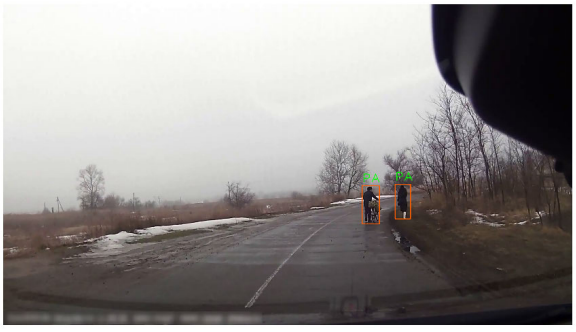
(b) Results of pedestrian detection using PPC, PC, PAC, PA action classification.

(c) Results of PPC, PC, PAC, PA detection



(d) Results of PPC, PC, PAC, PA detection.

(e) Results of PPC, PC, PAC, PA detection



(f) Results of PPC, PC, PAC, PA detection.

FIGURE 6. Example of pedestrian actions detection using a different approach.

differences between the predicted time to cross values and the observed ones, which is the common estimator evaluation metric (deviation of the prediction errors).

$$RMSE = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{\sum_{i=1}^{n}(P_i - R_i)^2}{n}}. \quad (1)$$

In this equation (1), P represents the Predicted value (predicted value), R the Actual values (real value), m represents the number of videos n represents the number of frames on the testing set.

We calculate the Confidence Interval (CI) to evaluate whether one model is statistically better than another one.

$$CI = 1.96\sqrt{\frac{P(100-P)}{N}}\%. \quad (2)$$

In this formulation, P represents the performance system (e.g., AP, mAP) and N represents the number of testing pedestrian samples.

V. RESULTS

The experiments were performed on the Jaad data set using the original video size. We independently provide the results for the pedestrian detection component followed by the

TABLE 2. The estimation of time to cross methods, independently of the detection-classification component. PPC:Pedestrian is Preparing to Cross the street. Real values: Testing, independently the pedestrian time to cross estimation on the all real pedestrian samples; Detected Values: Testing the detection component connected with the prediction component (Time to cross).

Learned on		Tested On	Past Time Steps	RMSE%	
				Real BB	Detected BB
Only PPC BB Coordinates	With Action Tag	All Samples	5	12.17	13.12
			14	9.36	11.72
			40	10.43	10.43
	Without Action Tag	All Samples	5	9.61	11.21
			14	13.38	13.34
			40	11.64	11.57
Only PPC BB Coordinates	with Action Tag	Only PPC BB Coordinates	5	5.87	8.03
			14	5.04	7.30
			40	4.76	4.88
	Without Action Tag	Only PPC BB Coordinates	5	5.75	7.14
			14	5.47	8.44
			40	5.86	8.26
All BB Coordinates	With Action Tag	All BB Coordinates	5	6.22	6.89
			14	5.57	8.71
			40	4.10	6.07
	Without Action Tag	All BB Coordinates	5	6.20	6.86
			14	5.36	6.32
			40	4.01	4.60

pedestrian detection with action classification capability, and finally, we present the performances for the estimation of the time to cross methods.

A. EVALUATION OF PEDESTRIAN DETECTION COMPONENT

In order to test the detection performance, we carried out several experiments. In our first detection experiment, we investigated the performances of RetinaNet [2] using the classical approach without action recognition on the JAAD data set. Our detection results are summarized in Table 1, where we show 56.05% mAP.

The detection performance connected with action recognition (using pedestrian action tags, PPC, PC, PAC, PA) of 26.36% mAP is presented in Tab 1.

This result is less than the previews one since it has to distinguish not only the pedestrian among other road users but also its actions. We deem that the four pedestrian action classes are quite difficult or even impossible to distinguish without environmental traffic context (crosswalk, sidewalk).

We observed the detection performance when pedestrian is crossing the street (PC) is the height one (65.57% AP), followed by the pedestrian is preparing to cross the street (PPC) (17.67% AP), the pedestrian is about to cross the street (PAC) (13.00% AP), and finally pedestrian intention is ambiguous (PA) (9.22% AP) (see Tab 1 and Fig 6 part 6a and 6b).

The second approach (using multiple pedestrian tags), although it detects the pedestrians, cannot be associated with the first method because it also instantly classify the action of the pedestrians during the detection step. We consider the second approach as a challenging one for CNN since four labels are used (PC, PCC, PAC, PA). This task, some time is challenging even for the human being (see Fig 6). The performance estimation is also biased, considering labels are ambiguous because some of them are very close or even belongs to two classes. Therefore its performance is less than the first detection approach.

B. EVALUATION OF PEDESTRIAN TIME-TO-CROSS COMPONENT

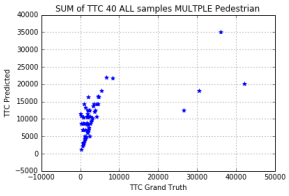
In Table 2, we present a comparison between our time to cross estimation models. We also present their performances on different prior time steps.

According to RMSE, the smallest error is the best one. We achieved 9.36% RMSE tested on real and 10.42% RMNS tested on detected values. This method was learned only on PPC samples with action tags and tested on all samples using 14 and respectively 40 frames as previous time step.

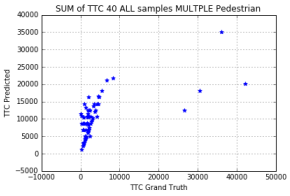
We obtained 4.76% RMSE tested on real values and 4.88% on detected values using only the PPC samples with action tags with 40 as previous time step.

For the method learned on all BB coordinates samples we achieved the best performance with 40 frames at a time step using only BB coordinates for both tested methods, (4.01% RMSE) real values and (4.60% RMSE) detected data methods.

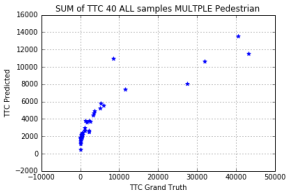
We observed the best for those methods where obtained with different time steps. We think this difference comes from the various length of the pedestrian sequences and complexity



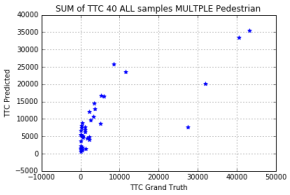
(a) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned only on PPC samples with action tags and tested on the all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



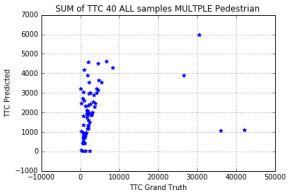
(b) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned only on PPC samples with action tags and tested on all samples: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



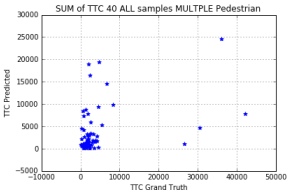
(c) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(d) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned and tested only on PPC samples with action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(e) TTC Predicted vs TTC Ground Truth real values on 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth



(f) TTC Predicted vs TTC Ground Truth detected values on 40 time step, learned and tested on all samples without action tags: Horizontal: SUM of time to cross predicted; Vertical: Sum of time to cross ground truth

FIGURE 7. **Performance of the pedestrian time to cross methods.**

of the data. However, the estimation of time to cross using all samples is more challenging for LTMS since it has to take into account even the pedestrian who is not prepared to cross the street, or whose intention is ambiguous.

In Figure 7, we plot the ground truth TTC versus predicted TTC on different time steps and for different approaches. We can observe that the estimation of TTC is globally satisfying. Indeed, the shape of the plot spread shows a roughly linear correlation between the real and the estimated values of TTC. It confirms that the TCC values can be directly estimated in a regression method using a deep learning approach.

VI. CONCLUSION

In this paper, we evaluated the estimation of the time to cross for pedestrians with deep learning approaches using the JAAD dataset.

We first studied different pedestrian actions to find out if a pedestrian is crossing the street, and based on this information, we estimate the time to cross for each detected pedestrian. We split the pedestrian Joint Attention for Autonomous Driving (JAAD) data set into four classes: pedestrian is preparing to cross the street (PCC), the pedestrian is crossing the street (PC), pedestrian is about to cross the street (PAC), and pedestrian intention is ambiguous (PA).

We evaluated the pedestrian detection approach, where all samples are tagged as pedestrian and not pedestrian and a pedestrian detection approach using multiple tags. The first method achieved better performance since it has only to distinguish the pedestrians from other road users in contrast to the second one which has also to recognize pedestrian actions. The second detection approach returned a weaker performance than the classical one.

The estimation of time to cross was learned using only PPC samples and respectively all samples. Since our global method is created in two stages, the first one could be applied whenever the pedestrian detector returns correctly the PPC event in contrast with the second one, which could be used without any restriction. The first one returns a better performance, but we consider the second one the more promising because it is more realistic, so we will continue to analyze it in our future our and also create an end-to-end detector-estimation time to cross approach.

REFERENCES

[1] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," Sep. 2016, *arXiv:1609.04741*. [Online]. Available: https://arxiv.org/abs/1609.04741

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017, *arXiv:1708.02002*. [Online]. Available: https://arxiv.org/abs/1708.02002

[3] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele, "Ten years of pedestrian detection, what have we learned?" in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2015, pp. 613–627.

[4] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" 2016, *arXiv:1602.01237*. [Online]. Available: https://arxiv.org/abs/1602.01237

[5] D. O. Pop, A. Rogozan, F. Nashashibi, and A. Bensrhair, "Incremental cross-modality deep learning for pedestrian recognition," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2017, pp. 523–528.

[6] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on YOLO network model," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Aug. 2018, pp. 1547–1551.

[7] M. Braun, S. Krebs, F. Flohr, and D. Gavrila, "EuroCity persons: A novel benchmark for person detection in traffic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1844–1861, Aug. 2019.

[8] M. Hoy, Z. Tu, K. Dang, and J. Dauwels "Learning to predict pedestrian intention via variational tracking networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3132–3137.

[9] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive Bayesian filters: A comparative study," in *Pattern Recognition*. Berlin, Germany: Springer, 2013, pp. 174–183.

[10] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 1–5.

[11] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 50–58.

[12] D. Ridel, E. Rehder, M. Lauer, C. Stiller, and D. Wolf, "A literature review on the prediction of pedestrian behavior in Urban scenarios," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3105–3112.

[13] Z. Fang and A. M. López, "Is the pedestrian going to cross? Answering by 2D pose estimation," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2018, pp. 1271–1276.
[14] R. Quintero, I. Parra, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction based on body language and action classification," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 679–684.
[15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. CVPR*, Jun. 2016, pp. 961–971.
[16] J. Hariyono and K.-H. Jo, "Pedestrian action recognition using motion type classification," in *Proc. IEEE 2nd Int. Conf. Cybern.*, Jun. 2015, pp. 129–132.
[17] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior," in *Proc. ICCV*, Oct. 2017, pp. 206–213.
[18] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 494–506, Apr. 2014.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: https://arxiv.org/abs/1512.03385
[20] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Understanding pedestrian behavior in complex traffic scenes," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 1, pp. 61–70, Mar. 2018.
[21] J. P. DeCruyenaere and H. M. Hafez, "A comparison between Kalman filters and recurrent neural networks," in *Proc. IJCNN Int. Joint Conf. Neural Netw.*, vol. 4, Jun. 1992, pp. 247–251.
[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[23] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," 2016, *arXiv:1605.06409*. [Online]. Available: https://arxiv.org/abs/1605.06409
[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," 2015, *arXiv:1512.02325*. [Online]. Available: https://arxiv.org/abs/1512.02325
[25] J. Redmon, S. Divvala, B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*. [Online]. Available: https://arxiv.org/abs/1506.02640
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and M. Blondel, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

DĂNUŢ OVIDIU POP received the M.Sc. degree in information technology from Petru Maior University, Târgu Mureş, Romania, in 2014. He is currently pursuing the Ph.D. degree with the INRIA Paris, RITS Team, Paris, France, in collaboration with the INSA Rouen - LITIS Laboratory, Normandie Université, Rouen, France, and the Department of Computer Science, Babeş University, Cluj-Napoca, Romania. His research interests include classification, detection, actions prediction, and tracking of road users based on vision, radar, and sensors fusion methods for the intelligent vehicle.

ALEXANDRINA ROGOZAN received the Ph.D. degree from Orsay University of Paris-Sud, in 1999. Since 2000, she has been an Associate Professor at INSA Rouen Normandie. She has published more than 70 articles in international journals, conferences, and book chapters. Her teaching activities include statistics and information theory. Her research interests include concern machine learning and multimodal fusion techniques in the area of intelligent vehicles, medical imaging, text-image-emotion understanding, and speech recognition, among others.

CLEMENT CHATELAIN received the Ph.D. from Rouen University, in 2006. Since 2007, he has been an Associate Professor at INSA Rouen Normandie. He has published more than 50 articles in the areas of machine learning, handwriting recognition, and medical imaging. His teaching activities include deep learning and signal processing. His research interests include machine learning and statistical pattern recognition, and mainly deep neural networks able to process mono and multidimensional sequences.

FAWZI NASHASHIBI received the master's degree in automation, industrial engineering and signal processing (LAAS/CNRS), the Ph.D. degree in robotics from Toulouse University, prepared in (LAAS/CNRS) Laboratory, and the HDR Diploma (Accreditation to research supervision) from the University of Pierre et Marie Curie (Paris 6). He is a Senior Researcher and has been the Program Manager of the RITS Team at INRIA (Paris-Rocquencourt), since 2010. Since 1994, he has also been a Lecturer at several universities (Mines ParisTech, Paris 8 Saint-Denis, Leonard de Vinci University - ESILV Professor, Telecom SudParis, INT Evry, and Ecole Centrale dElectronique) in the fields of image and signal processing, 3D perception, 3D infographics, mobile robotics, and C++/JAVA programming. He is the author of numerous publications and patents in the field of ITS and ADAS systems. His main research topics are in environment perception and multisensor fusion, vehicle positioning, and environment 3D modeling with main applications in intelligent transport systems and robotics. He is a member of the ITS Society and the Robotics & Automation Society. He is an Associate Editor of the several IEEE international conferences, such as IV, ITSC, and ICARCV, and journals.

ABDELAZIZ BENSRHAIR received the M.Sc. degree in electrical engineering and the Ph.D. in computer science from the University of Rouen, France, in 1989 and 1992, respectively. He was the Head of the Intelligent Transportation Systems Division, from 2007 to 2012, and a Co-Director of the Computer Science, Information Processing, and Systems Laboratory (LITIS), National Institute of Applied Science Rouen (INSAR), from 2002 to 2016, where he is currently a Professor with the Information Systems Architecture Department.