Universitat Autònoma de Barcelona

Behavior understanding of vulnerable road users by 2D pose estimation

Zhijie Fang

Behavior understanding of vulnerable road users by 2D pose estimation

A dissertation submitted by **Zhijie Fang** at Universitat Autònoma de Barcelonato fulfil the degree of **Doctor of Philosophy**.

Bellaterra, April 25, 2019

Co-Directors	**Dr. Antonio López Peña** Dept. Ciències de la Computació & Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr. David Vázquez Bermúdez** Element AI Montreal, Canada
Thesis committee	**Dr. Arturo de la Escalera Hueso** Dept. Ingeniería de Sistemas y Automática Universidad Carlos III de Madrid **Dr. Aura Hernández Sabaté** Dept. Ciències de la Computació Universitat Autònoma de Barcelona **Dr. Onay Urfalioglu** Dept. Automotive Engineering Lab Huawei Munich Research Center
International evaluators	**Dr. José Manuel Álvarez** AI-Infra NVIDIA, Santa Clara, CA, USA **Dr. Onay Urfalioglu** Dept. Automotive Engineering Lab Huawei Munich Research Center

This document was typeset by the author using LaTeX 2$_\varepsilon$.

ISBN: 978-84-945373-1-8

Printed by Ediciones Gráficas Rey, S.L.

Acknowledgements

Before I came to Spain, I was kind of scaring about the new life since that was my first time being out of China. After I arrived, people from Computer Vision Center and China (friends got the CSC scholarship) helped me a lot for getting used to the wonderful life here. Spanish friends were teaching me 'Hola, Guapa, Gracias/Gracies, Buenas Noches/Bona Nit' (I know, the latter cases are more important here).

Reading and writing papers in English are very difficult for me. Thanks a lot for the advising by my supervisor Antonio M. López during the whole PhD study. Also, Jiaolong Xu and my co-supervisor David Vázquez gave me a hand at the begging of my PhD. Here, I own Pras many thanks about the programming in Python and the English speaking (I still remember the Camino De Santiago trip with you).

Special thanks to the Chinese friends in CVC, namely HongXing, Xialei, Yulu, Yaxing, Lichao, Kanglei, Yangfei, Xiaoyi, Wangkai and Chenshen. Studying with all of you guys, I feel like I am in China sometimes. Also, Lidia and albert taught me lots of Spanish culture. I really appreciate the multiple culture here.

Finally, I would like to mention my dear girlfriend Anhong. After staying with her, I was more focusing on research. I started to have the idea of the paper which was presented in Sensor. Without the supports from my family, I cannot finish my PhD.

I know I forgot someone here. At the end, thanks all the people who help me during my PhD.

Abstract

Anticipating the intentions of vulnerable road users (VRUs) such as pedestrians and cyclists can be critical for performing safe and comfortable driving maneuvers. This is the case for human driving and, therefore, should be taken into account by systems providing any level of driving assistance, *i.e.* from advanced driver assistant systems (ADAS) to fully autonomous vehicles (AVs). In this PhD work, we show how the latest advances on monocular vision-based human pose estimation, *i.e.* those relying on deep Convolutional Neural Networks (CNNs), enable to recognize the intentions of such VRUs. In the case of cyclists, we assume that they follow the established traffic codes to indicate future left/right turns and stop maneuvers with arm signals. In the case of pedestrians, no indications can be assumed a priori. Instead, we hypothesize that the walking pattern of a pedestrian can allow us to determine if he/she has the intention of crossing the road in the path of the ego-vehicle, so that the ego-vehicle must maneuver accordingly (*e.g.* slowing down or stopping). In this PhD work, we show how the same methodology can be used for recognizing pedestrians and cyclists' intentions. For pedestrians, we perform experiments on the publicly available Daimler and JAAD datasets. For cyclists, we did not found an analogous dataset, therefore, we created our own one by acquiring and annotating corresponding video-sequences which we aim to share with the research community. Overall, the proposed pipeline provides new state-of-the-art results on the intention recognition of VRUs.

Resumen

Anticipar las intenciones de los usuarios vulnerables (VRU, por sus siglas en inglés), como peatones y ciclistas, puede ser crítico para una conducción segura y confortable. Este es el caso cuando conduce una persona y, por lo tanto, esas intenciones también deben ser tenidas en cuenta por los sistemas que brindan cualquier nivel de asistencia a la conducción, es decir, desde los sistemas avanzados de asistencia al conductor (ADAS, en inglés) hasta los vehículos totalmente autónomos (AVs, en inglés). En esta tesis doctoral, mostramos cómo los últimos avances en la estimación de la postura humana mediante visión monocular, es decir, aquellos que dependen de redes neuronales convolucionales (CNN, en inglés) profundas, permiten reconocer las intenciones de tales VRU. En el caso de los ciclistas, asumimos que siguen los códigos de tráfico establecidos para indicar, mediante señales con el brazo, futuros giros a la izquierda o derecha, así como la intención de pararse. En el caso de los peatones, no se puede suponer a priori ninguna indicación. En cambio, suponemos que el patrón de caminar de un peatón puede permitirnos determinar si él / ella tiene la intención de cruzar la carretera en el camino del vehículo (parcialmente) automatizado, de modo que este vehículo deba maniobrar en consecuencia (por ejemplo, reducir la velocidad o detenerse). En esta tesis doctoral, mostramos cómo se puede usar la misma metodología para reconocer las intenciones de los peatones y ciclistas. Para los peatones, realizamos experimentos con datos de Daimler y JAAD, disponibles públicamente. Para los ciclistas, no hemos encontrado datos análogos, por lo tanto, hemos creado nuestros propios datos mediante la adquisición y anotación de secuencias de video de ciclistas que pretendemos compartir con la comunidad científica. En conclusión, el método propuesto en esta tesis proporciona nuevos resultados de vanguardia en el reconocimiento de la intención de los VRU.

Resum

Anticipar les intencions dels usuaris vulnerables (VRU, per les sigles en anglès), com a vianants i ciclistes, pot ser crític per a una conducció segura i confortable. Aquest és el cas quan condueix una persona i, per tant, aquestes intencions també s'han de tenir en compte pels sistemes que brinden qualsevol nivell d'assistència a la conducció, és a dir, des dels sistemes avançats d'assistència al conductor (ADAS, en anglès) fins als vehicles totalment autònoms (AVs, en anglès). En aquesta tesi doctoral, mostrem com els últims avenços en l'estimació de la postura humana mitjançant visió monocular, és a dir, aquells que depenen de xarxes neuronals convolucionals (CNN, en anglès) profundes, permeten reconèixer les intencions de tals VRU. En el cas dels ciclistes, assumim que segueixen els codis de trànsit establerts per indicar, mitjançant senyals amb el braç, futurs girs a l'esquerra o la dreta, així com la intenció de parar-se. En el cas dels vianants, no es pot suposar a priori cap indicació. En canvi, suposem que el patró de caminar d'un vianant pot permetre determinar si ell / ella té la intenció de creuar la carretera al camí del vehicle (parcialment) automatitzat, de manera que aquest vehicle hagi de maniobrar en conseqüència (per exemple, reduir la velocitat o aturar-se). En aquesta tesi doctoral, mostrem com es pot fer servir la mateixa metodologia per reconèixer les intencions dels vianants i ciclistes. Per als vianants, vam realitzar experiments amb dades de Daimler i JAAD, disponibles públicament. Per als ciclistes, no hem trobat dades anàlogues, per tant, hem creat les nostres pròpies dades mitjançant l'adquisició i anotació de seqüències de vídeo de ciclistes que pretenem compartir amb la comunitat científica. En conclusió, el mètode proposat en aquesta tesi proporciona nous resultats d'avantguarda en el reconeixement de la intenció dels VRU.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and challenges

Automobiles play an important role in people's living. However, only in 2016 traffic accidents were the cause of 1.3 million death [43]. Among those death, pedestrians account for 23 percent and cyclists for 3 percent, as shown in Fig. 1.1. Fig. 1.2 shows that the number of death was increasing from 2000 to 2016. Therefore, it is urgent to develop driving assistance systems to protect these Vulnerable Road Users (VRUs; i.e. pedestrians and cyclists).

Researches in academia, automotive and technological companies develop advanced driver assistance systems (ADAS) and AI drivers to reduce accidents [17]. ADAS, which are active systems, include functions like adaptive cruise control, collision avoidance and pedestrian crash avoidance. These features can increase the safety of drivers or road users like cyclists and pedestrians. For example, when the ADAS detect a possible collision between the vehicle and a VRU, it can give a warning or even perform a stopping. Also, more advanced AI drivers, which are the brain of autonomous vehicles, can increase the safety of the pedestrians [64].

In autonomous vehicles, an AI drives by taking control actions after understanding the observations of the sensors. Perception is one of the key component of the system [36]. Computer vision based methods try to imitate human perception. Therefore, we need to research which kind of information is essential for human drivers. In particular, humans can predict the intentions of VRUs by their observation. Thus, a key functionality for AI drivers and ADAS is to anticipate the intentions of pedestrians and cyclists before the intended actions happen. The focus so far has been on already performed actions. Fig. 1.3 can explain the idea by comparing the differences between detecting an action vs detecting the intention of performing the action. There is a VRU approaching perpendicularly the future trajectory of the vehicle. Traditional automated systems do the vehicle path planning based on detecting the locations of VRUs. D-det is the detection based distance when the VRU is in the future vehicle trajectory. However, knowing the intention of the VRU as soon as possible provides the reaction distance of D-int, which is larger than

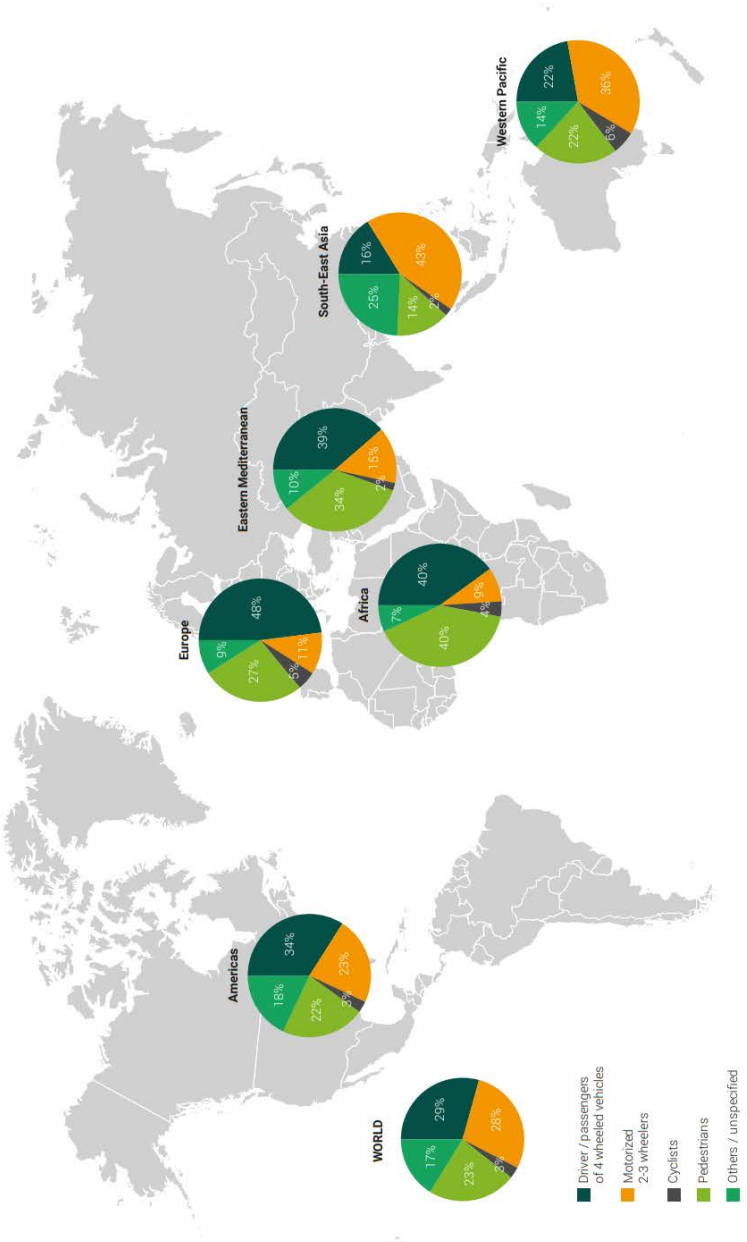


Figure 1.1 – Distribution of deaths by road user type according to WHO Region [43].

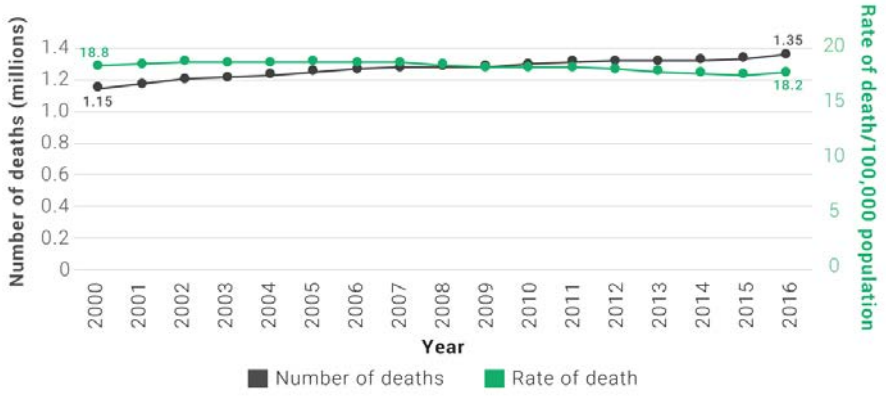


Figure 1.2 – Road traffic death per 100,000 population: 2000-2016 [43].

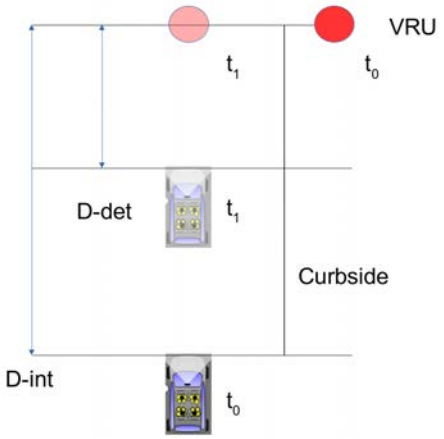


Figure 1.3 – Pedestrian intention detection can benefit automated systems in terms of reaction time and distance. Curbside is the frontier between the road (right) and the sidewalk (left). D-int refers to the distance to react for a system that can detect the VRU intention of crossing or not. D-det represents the distance to react of a system that only detects performed actions; in this case the VRU is already crossing the road without stopping in the curbside..

D-det, allowing an AI driver to slow down or even stop the vehicle earlier in a smoother way. Another case is when the VRU is going to stop at the sidewalk. Then,

if an AI driver can detect the stopping intention of the VRU, the vehicle can keep going without stopping.

To detect if a pedestrian is going to cross the road in front of the ego-vehicle is quite challenging. Because pedestrians can change their behaviors abruptly; *i.e.* the pedestrian can be in the sidewalk approaching the vehicle from walking to stopping, stopped to starting, etc. This is based on the observation of Daimler dataset [55] and Joint Attention for Autonomous Driving (JAAD) [46]. Recognizing the motion intentions of cyclists is also highly relevant since many times the ego-vehicle will need to overtake them. While we cannot assume that pedestrians will explicitly indicate their intentions, in the case of cyclists we can exploit traffic rules. In particular, as illustrated in Fig. 1.4, cyclists must indicate future left/right turns and stop maneuvers with arm signals. However, there is not almost literature for cyclist intention recognition, not even datasets.

The VRU intention recognition normally includes detection, tracking and intention recognition based on stereo cameras [55] or LIDAR [61]. Regarding the VRU detection and tracking, modern convolutional neural networks (CNNs) [51, 65] can perform these tasks with a high accuracy. Therefore, the challenge would be how to interpret the tracked detections correctly. In [53], a psychological study aims to find the most relevant information when human recognizes the pedestrian crossing behaviors. As shown in Fig. 1.5, we can see that pedestrian characteristics, dynamics and legs account for half of the pedestrian intention recognition. Therefore, some researchers were trying to obtain the relevant features by computing the optical flow (OF) [29] or silhouette changes [31] from stereo data . However, both of them need ego-compensation for the system. It is argued that such a compensation would need to be too precise to preserve small pedestrian movements (i.e., more precise than for tracking), which are crucial for recognizing intentions. In other words, the kind of procedures most common in the literature like relying on OF and depth, may not be as reliable as required to work "in the wild". In this context, the results found at [53] match the research of human activity recognition in [28], which shows how pose estimation is promising for general human activity recognition. Pose features are compared with OF, HOG and trajectories features for general human activity recognition task. It shows that pose features outperform all the others in such task [28]. Therefore, it is reasonable to assess the potential of pose estimation for detecting VRU intentions. Inspired by the state-of-the-art on 2D pose estimation [3], in this thesis, we hypothesize that 2D pose estimation can be, indeed, key for predicting intentions of VRUs.

(a) Turning Left

(b) Turning Right

(c) Another Turning Right

(d) Stopping

Figure 1.4 – Cyclist arm signals

1.2 Related work

1.2.1 Detection and tracking

Object detection and tracking are fundamental components for many applications; *i.e.* robotics, ADAS, autonomous driving and surveillance. Detection and tracking is the first step towards answering complicated questions like VRU intentions.

VRU detection is difficult to solve due to the wide variations in VRU appearance, weather conditions, lighting condition and scenarios. Examples based on computer vision can be seen in [8, 11, 15, 17, 66]. There are several important stages such as determining regions of interests (ROI), feature extraction and classification.

Regarding the ROI generation, pyramidal sliding window was one of most com-

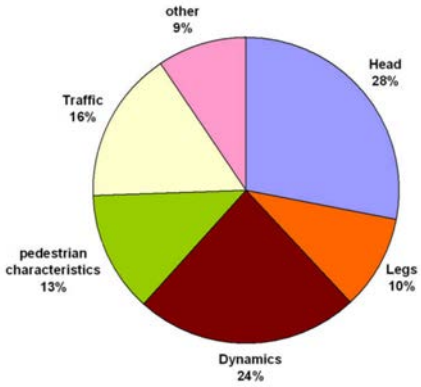


Figure 1.5 – Features used by human drivers for determining if a pedestrians is going to cross from a sidewalk to a road. Head: has actions of focusing, look left or right; Legs: already lifted foot for first step on the road or not; Dynamics: very briskly walking, does not move at all, straightway without deceleration; Pedestrian characteristics: upper body movement, distance to curb, age (like children are unpredictable); Traffic: traffic density, velocity of the vehicles; Other: comprises remaining categories such as zebra crossing, mother with child or group behavior [53].

mon approaches [44] due to its simplicity. Selective search [60], based on trained segmentations, was proposed as more efficient ROI generation method. Regarding feature extraction and classification, Histogram of Oriented Gradients (HOG) generated discriminative hand-crafted features robust to different scales and illuminations [7], capturing the shape and appearance of object, which combined with shallow classifiers such as SVM or Random Forest (RF) provided unprecedent detection performance at that time. Boosted classifiers can select better features to improve the detection accuracy [9]. The Deformable Part-based Model (DPM) can learn the different importance of object parts [12]. After the boost of performance in ImageNet competition in 2012 [34], the CNNs which can learn features and classifiers together in back propagation, became the state of the art on object detection by far. In fact, R-CNN [20] used CNNs to improve the performance of typical HOG/linear-SVM based methods. In [25], spatial pyramid pooling (SPP) was introduced to skip the image resizing which is used in R-CNN. Followed by the R-CNN and SPP, Fast-RCNN introduces ROI polling to accelerate the detection [19]. All the CNN-based methods (R-CNN/SPP/Fast-RCNN) use the selective search for the region proposals, which is not efficient. In order to avoid this problem, region

(a) Pedestrian pose estimation by [3].

(b) Cyclist pose estimation by [3].

Figure 1.6 – Examples of VRUs pose estimation.

proposal networks are trained with the object classification that makes the training and testing faster in Faster R-CNN [51]. However, still, the R-CNN methods use a region proposal process which is slower than one stage approaches like Single Shot Detector (SSD) [40] or You Only Look Once (YOLO) [48]. SSD and YOLO achieve real time with an accuracy similar to R-CNN based methods.

Regarding multiple object tracking (MOT), tracking-by-detection is the state of the art paradigm after the progress of object detection. Following this type of framework, offline methods such as [59] based on batch processing has the best performance recently. However, it cannot be adopted in online scenarios such as for autonomous driving vehicles. Deep simple online and realtime tracking (Deep-SORT) [65] used Kalman filtering and data association (CNN-based) to achieve the state of the art in online methods.

1.2.2 Activity recognition

Activity recognition has been studied for determining human actions in offline videos. Compared with object detection, video based activity recognition is more difficult. There are several main architectures for activity recognition. Either using RGB images or combined RGB with pre-computed OF as input to activity recog-

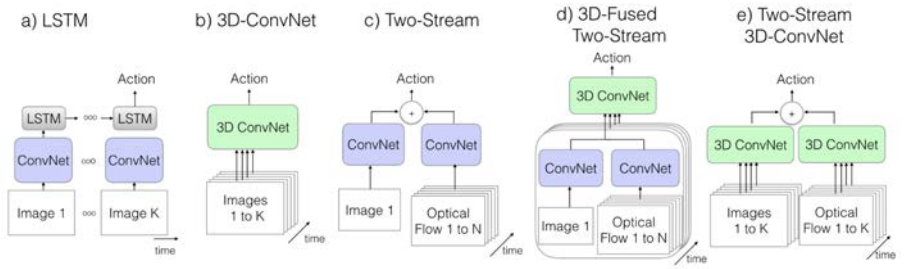


Figure 1.7 – Popular architectures for activity recognition [4].

nition models. For 2D CNNs, the information across different frames should be considered in recognition models. This information can be obtained by feature aggregation or recurrent neural networks (like LSTM [27]). Fig 1.7 shows several popular architectures for activity recognition. These kind of architectures usually take the whole video as input to detect the activities, and one single activity is supposed to happen in the video.

Focusing on the human itself, [28] studies which cues are import for human activity recognition. A subset of HMDB51 called J-HMDB [28] was built with new information of pose by using the puppet model [68]. Low level cues, such as dense trajectories (DT) on top of features such as HOG and HOF (histogram of OF) [62], are compared with high level cues of ground truth pose using J-HMDB. In all, pose related features showed most promising results in J-HMDB.

The pose information used in [28] comes from the ground truth. In real world, we need a pose estimation algorithm to obtain the pose. Pose models come from tree-structured graphical models [13], or non-tree models [63] which can handle symmetry and occlusion. CNN based models has boosted the performance of pose estimation significantly [3, 24]. Top-down [24] and bottom-up [3] are the two main strategies in pose estimation. In the former, humans are detected first, then pose estimation is performed. The later, based on Part Affinity Fields (PAFs), can even capture the spatial dependencies across different people. In all these methods, estimating the pose in low resolution is problematic. In [3] a head map of low resolution is used to obtain the pose. Instead, [42] estimates posterior probability maps as Gaussian mixture model for obtaining the pose. Pedestrian pose estimation in low resolution is improved largely compared with [3]. Fig 1.8 shows an example of pose estimation by skeleton fitting.

Finally, a very import clarification is that the activity recognition methods men-

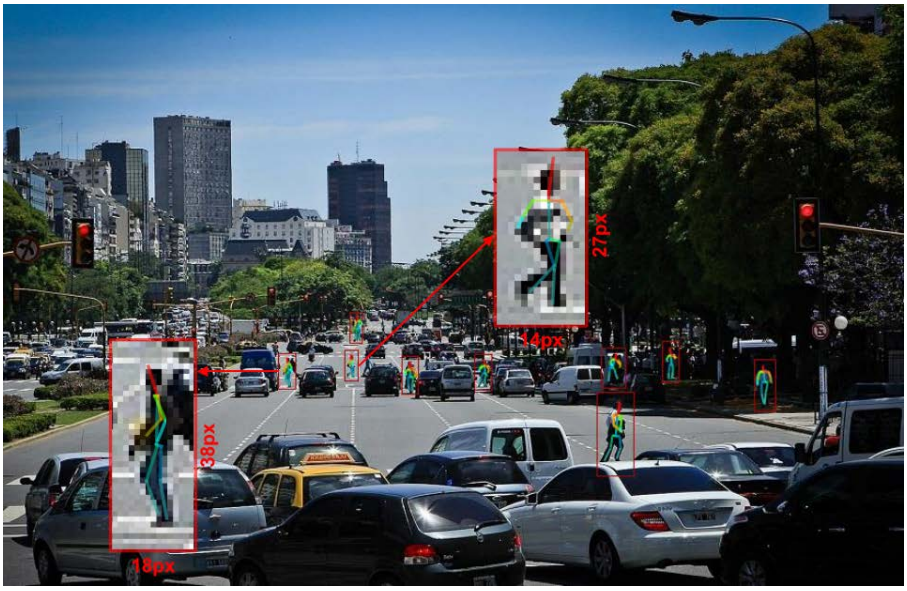


Figure 1.8 – Tiny pedestrian pose estimation [42].

tioned before use all the video (past and future) to classify the unique action in each video. In this PhD thesis, we can only use past frames to determine the intention of VRUs at current frame, we have no access to the future.

1.2.3 Intention recognition

There are relatively little research on the pedestrian intention recognition, all very recent [14, 29, 30, 31, 35, 46, 49, 54, 55, 56, 61]. Predicting pedestrian future is considered as pedestrian path prediction in [55]. This kind of prediction relies on the pedestrian dynamic models which convey location, speed and acceleration of the pedestrian. It uses a HOG/Linear-SVM based detector [7] running on dense stereo images to obtain the desired variables. The dynamic models with vehicle ego-motion compensation can predict the pedestrian future path (<2s) by using Interacting Multiple Model based on Kalman Filters (IMM-KF). In [29], Gaussian processes dynamical models and a probabilistic hierarchical trajectory matching improved the results. Moreover, it addresses the crossing *vs* stopping question (that we will discussed in Chapter 2).

HOG features based on the silhouette of pedestrian across several frames are

used to determine the intention of pedestrians [31]. Also, a 360° LIDAR based method is proposed in [61]. Head and body orientation are also studied for intention detection, relying on monocular [49] or stereo [14, 56] vision systems. Recently, [46] proposed a new dataset called JAAD which provides the pedestrian crossing tags and other attributes not included in previous publicly available naturalistic datasets.

Compared to pedestrian intention recognition, recognizing cyclist arm signals has received less attention so far. One core reason may be the lack of publicly available datasets for addressing this functionality. After [37, 38], it was publicly released one of the largest datasets focusing on cyclists, termed as Tsinghua-Daimler Cyclist Benchmark dataset (TDCB); however, acquired data and annotations are intended to support detection and orientation estimation tasks, but not cyclist arm signal recognition. In [1], the ground truth of TDCB was extended with wheel annotation for the case of bikes in side view; however, this is intended to support cyclist detection. Therefore, in this thesis, we introduce our Cyclist Arm Signal Recognition dataset (CASR).

Using a stereo camera setup, in [15, 32] it is detected whether the left arm of a cyclist observed from the back is up or down, which is used as a context cue within a path prediction module. However, an isolated accuracy analysis of such up/down arm classification is not performed. In order to perform such a classification, the disparity map computed from stereo image pairs is used to produce a binary mask of each detected cyclist, and template matching is applied to determine if the mask correlates with a left arm up or down. In particular, the scores of matching against multiple templates, the disparity values, and the image intensities, are used as core information to build a Naive Bayesian Classifier with uniform prior, which is responsible for the desired up/down arm classification.

1.3 Objectives

In this PhD we hypothesize that VRU intentions can be captured by 2D pose estimation, i.e. using a monocular vision system. Accordingly, our objective is to validate this hypothesis by peforming the following research:

1) Using off-the-shelf 2D object detection, tracking and pose estimation, show that pedestrian intentions can be predicted below half a second.

2) Using the same procedure than for detecting pedestrian intentions, detect cyclist arm signals, which indicate their maneuver intentions, below half a second.

Faster R-CNN and Mask R-CNN [24] are considered for the VRU detection. DeepSORT is used in the tracking part. For intention recognition, we use skeleton fitting [3], features based on the skeleton keypoints, and assess several methods such as

random forest, SVM and recurrent neural networks as classifiers [18]. Overall, ours is a monocular approach, *i.e.* it only relies on 2D image captures.

For the pedestrian intention recognition, two datasets are used in this PhD thesis. The first one was released by Daimler [55], and it is composed of short videos containing one person performing one action. These actions have been performed on purpose to build the dataset. The second one, called Joint Attention for Autonomous Driving (JAAD) [46], was recorded in naturalistic driving conditions. Using these two dataset we show the effectiveness of human pose estimation to recognize pedestrian intentions below half a second.

Regarding cyclist intentions, since there are not proper publicly available datasets, we build a new one called Cyclist Arm Signal Recognition dataset (CASR), which we consider also a contribution of this PhD. Using CASR and addional YouTube videos also annotated by us, we show how the same procedure used to recognize pedestrian intentions can be easily adapted to recognize cyclist arm signs with high accuracy.

1.4 Outline

The remaining of this thesis is organized as follows. Chapter 2 proposes skeleton based features for pedestrian intention recognition, using Daimler dataset for the first proof-of-concept. It shows that our method is the state-of-the-art for this task. In Chapter 3 we extend the analysis of our proposal by addressing the naturalistic scenarios provided by the JAAD dataset, which is done by integrating a full processing pipeline consisting on detection, tracking, pose estimation, and finally applying our intention recognition classifier. Chapter 4 presents our CASR dataset for cyclist arm signal recognition. Then, we show how 2D pose estimation can be used also as core functionality to perform cyclist arm signal recognition. These chapters are self-contained, including its own introduction, related work, methods, experiments, and final conclusions. Chapter 5 draws the global PhD conclusions and future work. Finally, Appendix A is included for completeness and giving some more details on *tools* (Faster R-CNN, Mask R-CNN, OpenPose, RF, etc) used along this PhD work.

2 On-board detection of pedestrian intention

Avoiding vehicle-to-pedestrian crashes is a critical requirement for nowadays advanced driver assistant systems (ADAS) and future self-driving vehicles. Accordingly, detecting pedestrians from raw sensor data has a history of more than 15 years of research, with vision playing a central role. During the last years, deep learning has boosted the accuracy of image-based pedestrian detectors. However, detection is just the first step towards answering the core question, namely *is the vehicle going to crash with a pedestrian provided preventive actions are not taken?* Therefore, knowing as soon as possible if a detected pedestrian has the *intention* of crossing the road ahead of the vehicle is essential for performing safe and comfortable maneuvers that prevent a crash. However, compared to pedestrian detection, there is relatively little literature on detecting pedestrian intentions. This paper aims to contribute along this line by presenting a new vision-based approach which analyzes the pose of a pedestrian along several frames to determine if he/she is going to enter the road or not. We present experiments showing 750ms of anticipation for pedestrians crossing the road, which at a typical urban driving speed of 50Km/h can provide 15 additional meters (compared to a pure pedestrian detector) for vehicle automatic reactions or to warn the driver. Moreover, in contrast with state-of-the-art methods, our approach is monocular, neither requiring stereo nor optical flow information.

2.1 Introduction

Avoiding vehicle-to-pedestrian crashes is a critical requirement for nowadays advanced driver assistant systems (ADAS) and future self-driving vehicles. Accordingly, detecting pedestrians from raw sensor data has a history of more than 15 years of research, with vision playing a central role [17]. During the last years, deep learning has boosted the accuracy of image-based pedestrian detectors [50]. However, detecting the pedestrians is just an intermediate step since the question to answer is if the ego-vehicle is going to crash with a pedestrian provided preventive actions are

not taken. For instance, using Fig. 2.1 Left) as reference, a pure pedestrian detection approach would report that a pedestrian may be in danger as a function of his/her location with respect to the road ahead of the ego-vehicle, his/her distance to the vehicle, and the vehicle motion (direction and speed). However, knowing as soon as possible if a detected pedestrian has the *intention* of intersecting the ego-vehicle path (expecting the vehicle slowing down or braking) is essential for performing safe and comfortable maneuvers preventing a crash, as well as having vehicles showing a more respectful behavior with pedestrians (see [16, *Challenges*]).

Despite the relevance of detecting pedestrian intentions, since pedestrian detection is the first hard task to solve, most of existing literature focuses on the latter topic as can be seen in the surveys [8, 11, 17, 66], and relatively little on the former one [14, 29, 30, 31, 35, 46, 49, 54, 55, 56, 61]. This paper aims at contributing in this line by presenting a new vision-based approach which analyzes the pose of a pedestrian along several frames to determine if he/she is going to enter a road area that may generate a risk of crashing. The presented method relies on: (a) a CNN-based pose estimation method that detects pedestrians and provides their skeleton simultaneously [3]; (b) a fast classifier based on a set of high-level features extracted from a detected skeleton and a normalized SVM that processes them. The literature of action recognition in videos[1] supports the hypothesis that high-level features (*e.g.* skeleton joints) are more action-informative than low-level ones ({*e.g.*} HOG, HOF) [28]. In addition, since the pose estimation method is a single-frame monocular approach, in contrast with state-of-the-art methods for detecting pedestrian intentions, ours neither requires stereo nor optical flow information.

For the present study, we rely on a publicly available dataset designed to assess methods for detecting pedestrian intentions [55]. In this dataset, it is considered that a pedestrian enters in a risk area when he/she moves from the sidewalk towards the road ahead of the ego-vehicle, as seen in Fig. 2.1 Right). We present experiments showing 750ms of anticipation for pedestrians crossing the road, which at a typical urban driving speed of 50Km/h can provide 15 additional meters (compared to a pure pedestrian detector) for vehicle automatic reactions or to warn the driver. At the same speed, initiating emergency brake with 160ms of anticipation over a 660ms time to collision can reduce the chance of injury requiring hospitalization from 50% to 35% [41].

[1] Just a technical but important clarification for the general reader. For detecting pedestrian intentions we have to take per-frame decisions, we are allow to use past frames but not future frames; while in video-based action recognition the full video is used (past, present and future).

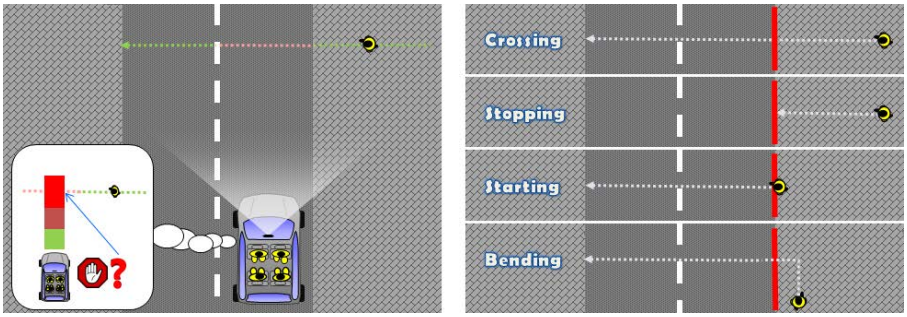


Figure 2.1 – *Left)* Anticipating as much as possible the intentions of a pedestrian allows for safer and more comfortable maneuvers. For instance, we would like to know if the pedestrian is going to enter the road while walking towards it from the sidewalk; or in general if it is going to enter a critical area that the ego-vehicle can compute as its predicted driving path. *Right)* Different situations taking the curbside (red line) as reference [56]. From top to bottom: a pedestrian will be *crossing* the road without stopping; a pedestrian walking towards the road will be *stopping* at the curbside; a pedestrian that was stopped at the curbside is *starting* to walk for entering the road; a pedestrian walking parallel to the curbside (parallel to the trajectory of the ego-vehicle) will be *bending* towards the road. Here we plot the pedestrian walking away from the ego-vehicle, but walking towards the ego-vehicle and bending would fall in the same category.

2.2 Related Work

One of the first attempts of predicting pedestrian *future* is more related to pedestrian path prediction, *i.e.* without an explicit step for determining the intentions of the pedestrians [55]. Pedestrian dynamic models are proposed conveying location, speed and acceleration. The measurements to set such variables come from a HOG/Linear-SVM based pedestrian detector [7] operating on dense stereo images at 16 fps. An Interacting Multiple Model based on Kalman Filters (IMM-KF) is used to predict the future path (<2s) of a pedestrian according to the used dynamic model and vehicle ego-motion compensation. Overall a simple constant speed velocity model (with white noise acceleration) was on par with more sophisticated models. In a following work [29], results are improved by considering Gaussian process dynamical models and a probabilistic hierarchical trajectory matching (involving particle filters, PCA and mean-shift). In this case, not only stereo data is used but the dynamical models also rely on motion features extracted from dense optical

flow with vehicle ego-motion compensation. Intuitively, the method implicitly tries to predict how the silhouette of a tracked pedestrian evolves over time. Moreover, it explicitly assessed the question of whether a pedestrian will cross from the side walk to the road ahead of the ego-vehicle, *i.e.* crossing *vs* stopping in Fig. 2.1 Right). For doing that, trajectories of the *stopping* and *crossing* classes are learned and, then, unobserved testing trajectories are classified according to the trajectory matching method.

In this paper we present an explicit data-driven model to detect pedestrian intentions using skeleton features, which are used without requiring to individually track them. In fact, tracking is only assumed for a pedestrian as a whole, which is unavoidable for any method aiming at detecting intentions. Our proposal obtains equivalent results to [29] in the *crossing vs stopping* classification, being much simpler and only relying on monocular information, neither on dense stereo as in [29, 55], nor on dense optical flow with ego-motion compensation as in [29].

In [31], a stereo-vision system is also used to assess the silhouette of the pedestrians for determining their intentions (other authors used 360° LIDAR [61]). The proposed method has the advantage over previous ones of requiring vehicle ego-motion compensation only for tracking of the pedestrians, but not for computing features for detecting intentions. It is argued that such a compensation would need to be too precise to preserve small pedestrian movements (*i.e.* more precise than for tracking), which are crucial for recognizing intentions. As in [31], the method that we present here does not require ego-motion compensation by itself (only if the tracking uses it). Moreover, our results are comparable (in fact, slightly better) to [31] without requiring dense stereo.

Other approaches focus on on-board head and body orientation estimation as a cue for detecting the intention of a pedestrian, from monocular [49] or stereo [14, 56] images with vehicle ego-motion compensation. However, it is unclear how we actually can use these orientations to provide intention estimation, neither how much additional time this information can bring to perform a reactive maneuver. Indeed, for a time to collision below 2s, pedestrians tend to look at the vehicle before crossing [46]. However, we are not aware of any work reporting with how much anticipation this happens; for instance, in [46] pedestrian behavior statistics are based on observations at the point of crossing (e.g. the curbside in Fig. 2.1 Right). In our proposal, we rely on a 2D pedestrian pose estimation method, therefore, we are already implicitly taking into account the kind of body orientation that works such as [14] try to compute; in fact, the one we use is more fine grained. The method used to obtain the pose also provides head orientation; however, it is not as robustly detected as the rest of the body. Thus, we consider head pose estimation as an additional cue we could consider in the future since it can complement our current study. On the other hand, the experiments reported in [56] suggest that

head detection is not useful for distinguishing *crossing vs stopping*, although it is for detecting *bending*.

In [46] it is suggested to further study the gait patterns of pedestrians, which is what our method actually do by using a data-driven approach. In fact, in [54] it is explicitly said that *a lack of information about the pedestrian's posture and body movement results in a delayed detection of the pedestrians changing their crossing intention*. Thus, our proposal of using a 2D pose estimator for analyzing intentions is aligned with these suggestions.

The rest of the paper is organized as follows. In Sect. 2.2 we summarize the works most related to this paper. In Sect. 2.3 we describe our approach for detecting pedestrian intentions. In Sect. 2.4 we present the performed experiments and discuss the obtained results. Finally, Sect. 2.5 draws the summary and future work.

2.3 Detecting Pedestrian Intentions

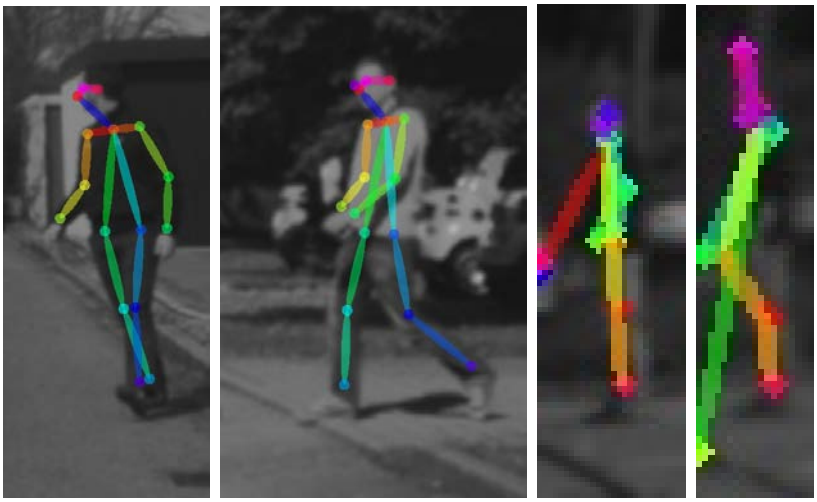
2.3.1 Our Proposal in a Nutshell

The proposed approach is summarized in Fig. 2.2. The first step consists of pedestrian detection and tracking, which is a common step to any method assessing pedestrian intentions. We are agnostic to the methods used for these tasks, we only assume that for each pedestrian we will have a 2D bounding box (BB) that comes from the combination of detection and tracking. The second step consists of the use of a 2D pose estimation method that results on the fitting of a skeleton model to the pedestrian contained in each BB. In this case, we propose the use of the recent method presented in [3]. It relies on a two-branch multi-stage CNN trained on the *Microsoft COCO 2016 keypoints challenge dataset* [39]. When applied to a BB containing a pedestrian, it is able to perform the skeleton fitting being robust to pedestrian shifts (because inaccuracies in the detection and tracking step) and scaling (because different pedestrian sizes and distance to the camera) within the BB. Fig. 2.3 shows different skeleton fittings as a function of the distance. The algorithm starts to fail only at large distances (*e.g.* 40m in the figure's example). The third step consists of extracting a feature vector, namely ψ, based on the skeleton fitted to each tracked pedestrian (Sect. 2.3.2). In fact, since intentions are shown as an action over time, for each tracked pedestrian, at frame t we concatenate the feature vectors of the last T frames, giving rise to a per-pedestrian feature vector $\Psi_t =< \psi_t, \psi_{t-1}, \ldots, \psi_{t-T} >$, where ψ_i stands for the feature vector at frame i. Fig. 2.4 shows skeleton fitting results for BBs coming from 10 consecutive frames ($T = 10$) depicting pedestrians performing the four situations we are considering in this paper. The final step consists of applying a classifier $\mathscr{C}$ on Ψ that fires for a

pedestrian intention we want to assess (Sect. 2.3.3).

Note that the proposed method does not explicitly require global egomotion compensation. The detection-tracking process is already sufficient to capture the pose evolution in which our method relies on. Therefore, explicit egomotion compensation would be required only if the tracking itself relies on it.

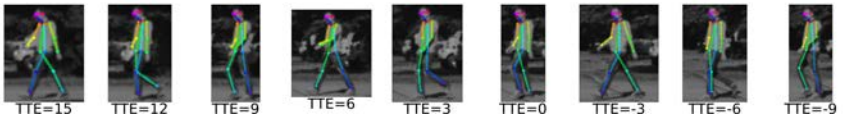
Figure 2.2 – Proposed method. Monocular frames are continuously acquired and processed for detecting and tracking pedestrians. For each tracked pedestrian our proposal consists of: estimating his/her 2D pose by skeleton fitting, computing features from the fitted skeleton; input them to a learned classifier which will output the intention of the pedestrian.



(a) 13m (b) 18m (c) 40m (d) 45m

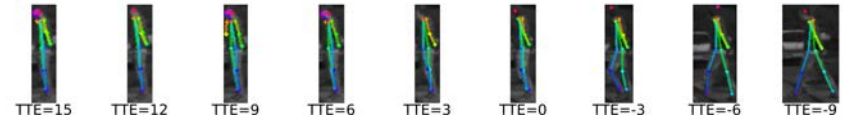
Figure 2.3 – 2D pose estimation, *i.e.* 2D skeleton fitting, at increasing pedestrian-vehicle distances.

(a) Stopping



(b) Crossing

(c) Bending



(d) Starting

Figure 2.4 – Skeleton fitting for the four situations considered in this paper. We show a sequence for each situation. TTE stands for *time to event*. TTE=0 is when the event of interest happens: stopping at the curbside, crossing the curbside, bending, and starting to walk from the curbside. Positive TTE values correspond to frames before the event, negative values to frames after the event.

2.3.2 Skeleton Features

In Fig. 2.5 we can see that the fitted skeleton is based on 18 keypoints. Note that left and right body parts are distinguished. However, not all keypoints are always located very accurately when processing on-board images. We found as most stable the 9 keypoints highlighted with a star, which correspond to the legs and to the shoulders. Note that these are highly relevant keypoints since ultimately the legs are executing the pedestrian intentions of continue/start walking or stopping; while having keypoints from shoulders and legs provides information about global body orientation.

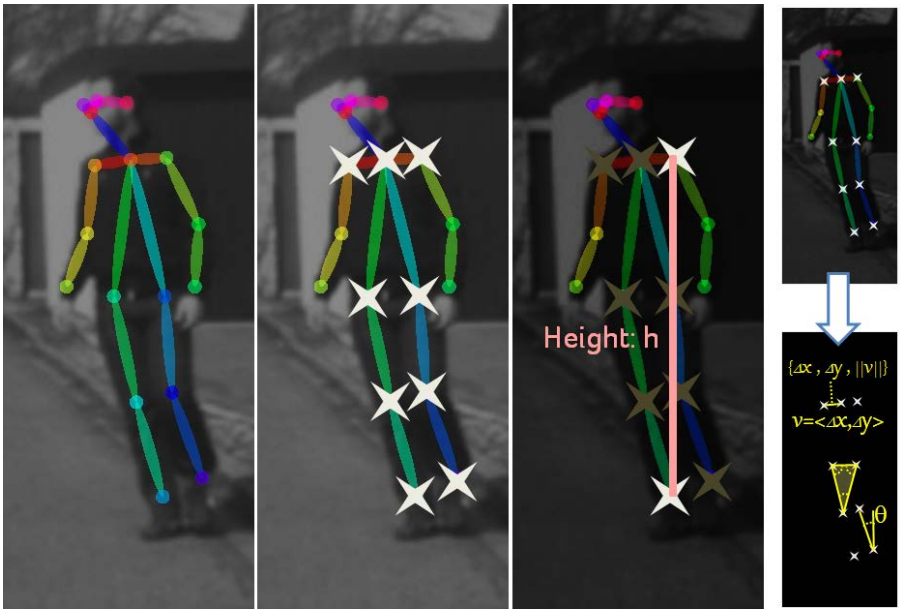


Figure 2.5 – Skeleton fitting is based on 18 keypoints, distinguishing left and right [3]. We use the 9 keypoints highlighted with stars. The upper keypoint among those and the lower are used to compute height h, which is used as scaling factor for normalizing the keypoint coordinates. Then, using the normalized keypoints, different features based on relative angles and distances are computed as features. For instance, to the right we see several examples: (1) distance in the x (column) and y (row) axes and Euclidean distance between two keypoints (Δx, Δy, $\|v\|$); (2) angle between two keypoints (θ); (3) the three angles of a triangle formed by three keypoints. After normalizing by h these seven values, they become components of the feature vector ψ_i of frame i. Computing similar values by taking into account all the keypoints we complete ψ_i.

From the selected keypoints, we compute features. First, we perform a normalization of keypoint coordinates according to a factor h defined as shown in Fig. 2.5, which is proportional to the pedestrian height. Then, different features (conveying redundant information) are computed by considering distances and relative angles between pairs of keypoints, as well as triangle angles induced by triplets of keypoints. In total we obtain 396 features (dimension of ψ). Since we concatenate the features collected during the las T frames, our feature vector Ψ has

dimension $396T$.

It is worth to remind that we know the position of any keypoint along the different frames because they correspond to an specific and unique anatomical part of the fitted skeleton. Thus, a priori it makes sense to account for keypoint time differences. In fact, we did it; however, results did not improve and thus we discarded across-frame features. We think the reason is that the proposed Ψ already conveys sufficient information to perform the further classification task.

2.3.3 Random forest or SVM Classifiers

In this paper we consider binary classifiers which rely on learned frontiers and output a normalized score. In particular, we tested the Random Forest (RF) and Support Vector Machine (SVM) methods. RF is able to learn non-linear frontiers and outputs a probability value. For the SVM we apply Platt scaling on RBF Kernel scores. We access all these functionalities by using scikit-learn [45].

Independently of using SVM or RF, following the literature evaluation protocols [31, 56], in this paper we assume a procedure for detecting pedestrian intentions which is based on the following binary classifiers:

- $\mathscr{C}_c$: Continue walking perpendicularly to the camera (~crossing) *vs* stopping.
- $\mathscr{C}_b$: Continue walking parallel to the camera *vs* bending.
- $\mathscr{C}_s$: Continue stopped *vs* starting to walk perpendicular to the camera.

Note that *Continue walking perpendicularly to the camera* is equivalent to *crossing* given a fiducial point of interest such a curbside or a frontier of risk determined by the ego-vehicle future motion.

Each classifier can have a threshold to determine if it fires or not. With a simple pedestrian tracking we may need to test all classifiers, while with a tracker that keeps proper pedestrian motion vectors, we may need to apply only one of those classifiers.

2.3.4 LSTM Classifiers

We have also explored the Long Short Term Memory networks (LSTM) [27] as an alternative to aggregate temporal information. We follow an implementation similar to [18]. Fig. 2.6 shows our intention recognition framework based on LSTMs.

Figure 2.6 – LSTM framework for intention recognition (10 frames as input).

Table 2.1 – Number of sequences of training and testing for each type of pedestrian intention [55].

	Stopping	**Crossing**	**Bending**	**Starting**
Training	9	9	12	5
Testing	8	9	11	4
Total	17	18	23	9
VehicleMoving	12	15	18	9
VehicleStanding	5	3	5	0

2.4 Experimental Results

2.4.1 Dataset

Unfortunately, at the moment of doing this research, the only publicly available dataset (to the best of our knowledge) with ground truth (GT) annotations for assessing pedestrian intentions is the one first introduced in [55] and recently used in [31, 56]. The dataset contains 68 sequences (9,135 frames in total) recorded on-board with a stereo camera (here we only use the left frame of each pair) placed in the windshield forward facing the road ahead. The images are taken at 16 FPS and their resolution is of 1176×640 pixels. Among the sequences, 55 were taken with vehicle speeds ranging from 20 to 30 Km/h, while for 13 the vehicle was standing. In order to make easier comparisons, the sequences are separated into training and testing as can be seen in Table 2.1. The pedestrians come with two types of BBs, namely manually provided GT BBs and BBs from a HOG/Linear-SVM classifier. Event tags are provided (crossing, stopping, bending, starting) as well as the *time to event* (TTE) in frames (Fig. 2.4).

2.4.2 Evaluation Protocol

Since we consider the same set of intentions as [31], we also use the same train-test partition of the working sequences (shown in Table 2.1). We also follow the recommendation of [31] to select positive and negative samples when training the classifiers; *i.e.* we divide a training sequence in three segments of samples: positives–not used—negatives. We will use the notation $A-B$, with $A>B$, meaning that frames with $TTE>A$ are used as positive samples, and frames with $TTE\leq B$ are used as negative samples; thus, frames with $TTE\in(B,A]$ are ignored during training.

As in [31, 56] we use plots of *intention probability vs* TTE. With this type of plot it is easy to see how many frames we can anticipate a pedestrian action (*e.g.* for crossing *vs* stopping), or how fast we can react to it (*e.g.* for starting and bending). Since there are several testing sequences per intention, mean and standard deviation are plotted. In addition, also following [31], we use these plots to select a proper probability threshold so that we can also present plots of what they call *accuracy vs* TTE. However, we prefer to call it *predictability*, *i.e.* for each TTE is given a normalized measurement of how feasible it is to detect the action under consideration at that TTE. This predictability measurement is computed as follows. First, since the testing sequences have different length, we align them by making their TTE=0 frame to coincide. Then, from the minimum TTE over all the sequences until the maximum TTE, we compute a predictability value for each TTE as follows. All the frames corresponding to the current TTE (*i.e.* coming from the different testing sequences) are considered. For each of those frames we apply our method given a classification threshold for the probability of the intention/action under consideration. Then, we divide the number of frames rightly classified by the number of total frames evaluated. Predictability zero indicates that we cannot detect the intention/action, while predictability one means that we can.

Again following [31, 56] we use both the GT pedestrian BBs as well as the detections provided by the HOG/Linear-SVM. Although human-provided BBs are not necessarily consistent, we can take them as the output of a state-of-the-art pedestrian detection and tracking system (nowadays it could rely on CNN-based models). The hyper-parameters of the classifiers are set here as the ones providing the best performance. For the SVM classifier, C was adjusted by starting in 1 and applying a factor of $\times 10$ until 10^6. C=10000 provided the best results. Small variations around this value did not provide significant better results. For the RF we tested different depths ranging from 7 to 29 in steps of 2, and using 100, 200, 300, and 400 trees. Finally, we selected 21 as depth and 300 trees. The HOG/Linear-SVM classifier is nowadays far from the state-of-the-art, but we use it for a proper comparison with [31, 56] in terms of pedestrian intentions. However, we have not implemented

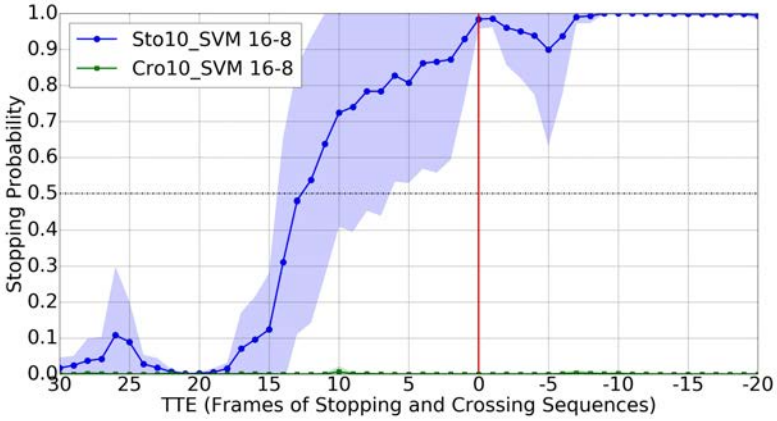
a tracker for extrapolating detections from previous frames to a frame where a pedestrian is missed by the HOG/Linear-SVM detector, the reason is that we have quantified these cases as ≈ 2%; thus, when this happens we take the corresponding GT BB and add a 10% noise to its defining coordinates (this noise level is used also in [29, 56] for perturbing GT BBs).

We have not worked on code optimization; thus, we are not including an in deep analysis of computation time. However, we can indicate several reference times. At testing time the pose estimation method runs at 10 frames per second in a consumer graded GPU (NVIDIA GeForce GTX-1080) [3]. Our non-optimized code, which uses estimated poses to predict pedestrian intentions, takes less than 15 ms in an INTEL Xeon E5-1620 v3 PC. Thus, the main computation time corresponds to pose estimation. In training time, given the already trained pose estimation CNN model, each of our classifiers for detecting pedestrian intentions is trained in approximately one hour.

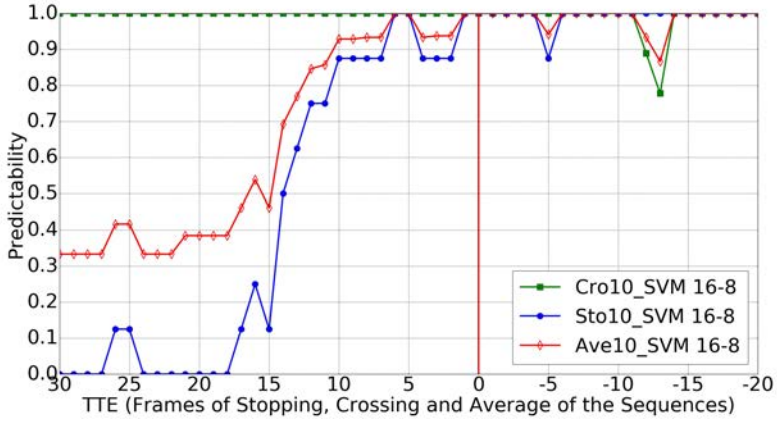
2.4.3 Crossing *vs* Stopping

In the sequences of the used dataset we can see that the walking cycle is of ≈ 10 frames; therefore, for developing $\mathscr{C}_c$ (Crossing *vs* Stopping) we started with a temporal sliding window of $T = 10$ as well as using a RBF-SVM frontier. We also set the best performing $A - B$ pair in [31], *i.e.* $16 - 8$. Fig. 2.7 shows the results of comparing the probabilities of crossing *vs* stopping for different TTE values, as well as the accuracy for a selected threshold; this case corresponds to the use of GT pedestrian BBs. Fig. 2.7a shows that when we apply $\mathscr{C}_c$ to the crossing sequences the probability values are almost zero with very low standard deviation; while when applied to stopping sequences, the probability starts to grow significantly in the TTE range of 15-10 (in these sequences TTE=16 corresponds to one second of anticipation). Thus, the classifier is very sure about when to stop, which is very important from the point of view of safety. By setting a probability threshold of 0.2 we can see in Fig. 2.7b that at TTE=12 we reach the 0.8 of average predictability. Note that TTE=12 are 750ms before the event, which is very interesting since in [29] it is reported that humans reach 0.8 predictability with less anticipation, namely 570ms. Thus, although a comprehensive human-*vs*-machine comparison is out of the scope of this paper, these evidences suggest that our prediction system may be on pair with humans for this task. Moreover, in Fig. 2.8 we can see that when using the BBs of a basic pedestrian detector (HOG/Linear-SVM hear) the results are very similar, also with TTE=12 for the 0.8 of predictability.

For the GT BBs case, [31] reports TTE=11 for the 0.8 of predictability, so our results are comparable but not requiring dense stereo. For the BBs coming from pedestrian detection, [31] reports TTE=8 for the 0.8 of predictability, while our

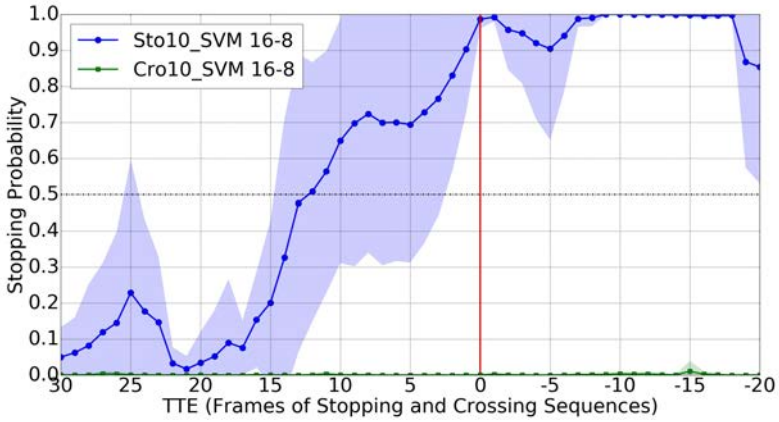


(a) Classification probability (mean as curves, standard deviation as colored areas).

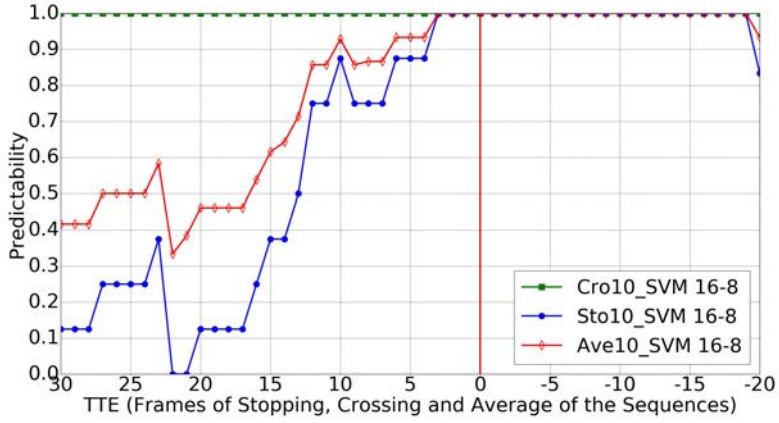


(b) Predictability for $\mathscr{C}_c$ with threshold 0.20.

Figure 2.7 – Results for the *crossing vs stopping* classification task ($\mathscr{C}_c$), using GT pedestrian BBs, a time sliding window of 10, the RBF-SVM classifier and 16 – 8 as trade off for setting positive and negative frames during training. 'Cro' curve means *applied to testing sequences of crossing*, 'Sto' curve means *applied to testing sequences of stopping*. Note that the frames from the stopping sequences are rightly classified if $\mathscr{C}_c > 0.20$, while for the crossing sequences those are the wrongly classified.



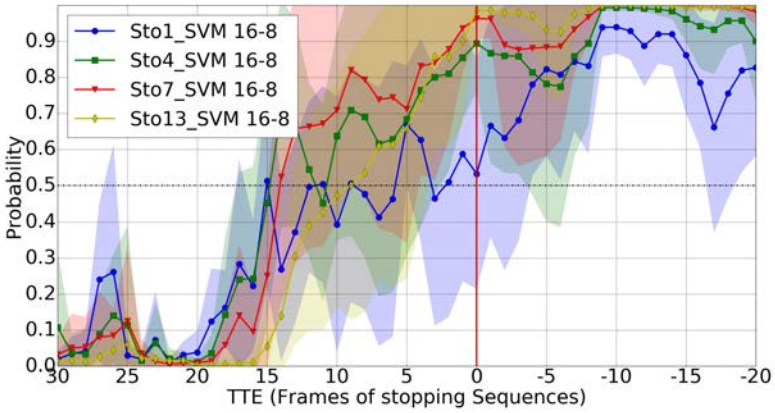
(a) Classification probability.



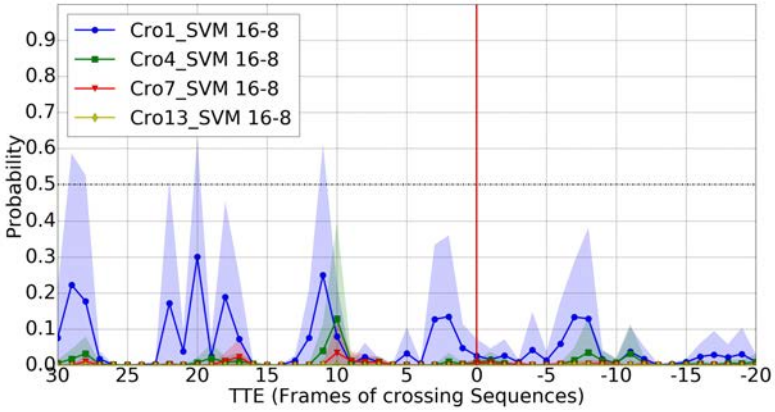
(b) Predictability for $\mathscr{C}_c > 0.20$.

Figure 2.8 – Analogous to Fig. 2.7, but using the BBs of the provided pedestrian detections.

method still reports TTE=12. We think, this is due to the fact that our proposal relies on higher level features (based on skeleton keypoints), an observation also reported on action recognition in videos [28]. Moreover, the used 2D pose estimation methods add shift invariance to the exact pedestrian location within the detection



(a) Stopping sequences.



(b) Crossing sequences

Figure 2.9 – Classification probability for several temporal sliding windows ($T \in \{1,4,7\}$) applied to stopping and crossing sequences.

BBs which use to come with inaccuracies. In addition, although it is difficult to report a direct comparison with [56] because accuracy is not reported, looking at the plot of stopping probability *vs* TTE for stopping scenarios, it seems that the method proposed in [56] is not robust; in fact, the authors themselves report that

head detection is not useful for this particular task (while it is for bending actions). In order to complement our study, we also checked the results when using different sizes of temporal sliding window; in particular, we also tested $T \in \{1,4,7,13\}$. Results can be seen in Fig. 2.9 when using GT BBs. Note how results improve as we increase T; however, these results are not as good as when using $T = 10$ as seen by comparison with Fig. 2.7. When using BBs coming from the HOG/Linear-SVM pedestrian detector, the results are analogous; thus, we do not plot them here for the sake of simplicity.

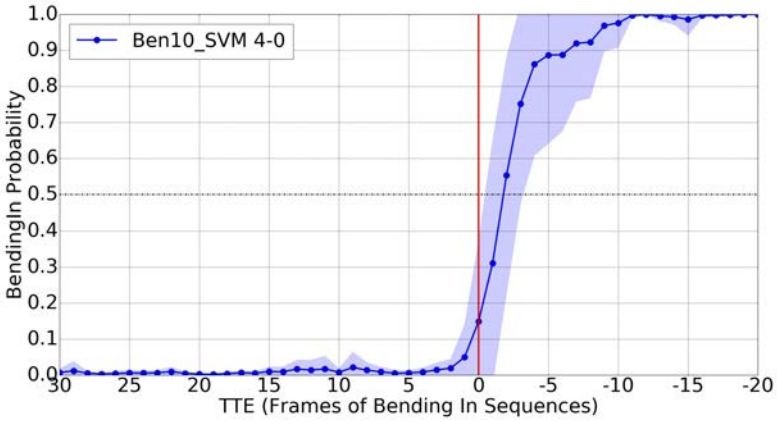
For these experiments we also used the RF method, however for achieving the 0.8 of predictability we have TTE=6; which is significantly worse. Obviously, this does not imply that RBF-SVM is better than RF in general, we only report the result we obtained for this task given the available training and testing sets.

2.4.4 Bending

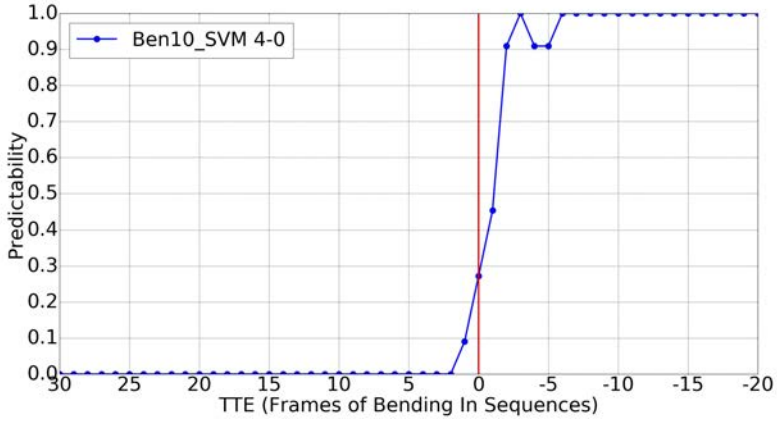
Following [31], for training $\mathscr{C}_b$, we set the $A-B$ pair as $4-0$. Again, we report only results for $T = 10$ and RBF-SVM since for $T \in \{1,4,7\}$ and RF they were worse. In this case, we would like to mention that rather than predicting the intention of bending, which is extremely difficult, the aim is to understand that this is happening as soon as possible.

In Fig. 2.10 we can see that for GT BBs we reach the 0.8 of predictability for TTE=-2, *i.e.* after 125ms of the event happening. In Fig. 2.11 we plot the analogous results using the BBs from the pedestrian detector. We see that before the action happens, the system outputs less stable probabilities. However, by using the proper threshold, we still reach 0.8 predictability for TTE=-4 (250ms). Note that [31] reports TTE=-4 when using GT BBs, and TTE=-5 for BBs from pedestrian detections (312ms).

We have visually inspected the result and found that for far pedestrians ($TTE >$ 10 since the vehicle is approaching the pedestrian in this case), the 2D pose estimation has difficulties in distinguishing back and front pedestrian views; which introduces an instability that induces differences in training an testing time. This is why in Fig. 2.11a the probabilities fluctuate more for $TTE > 10$. On the other hand, comparing to Fig. 2.10a it seems that at far distances by just having a more accurate pedestrian detector and so providing more accurate BBs, can already help the pose estimator. In any case, this back/front viewpoint confusion is a point for improvement in our future work. We think that for this particular action, head orientation can be also tested to assess if we can predict the action more closely to TTE=0.



(a) Classification probability.

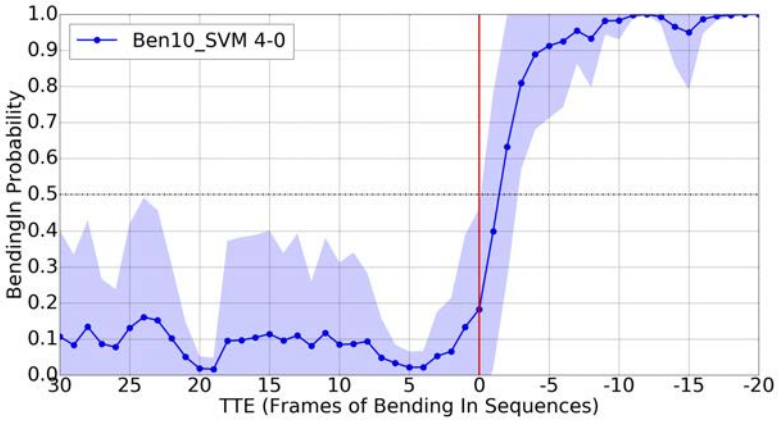


(b) Predictability for $\mathscr{C}_b > 0.16$.

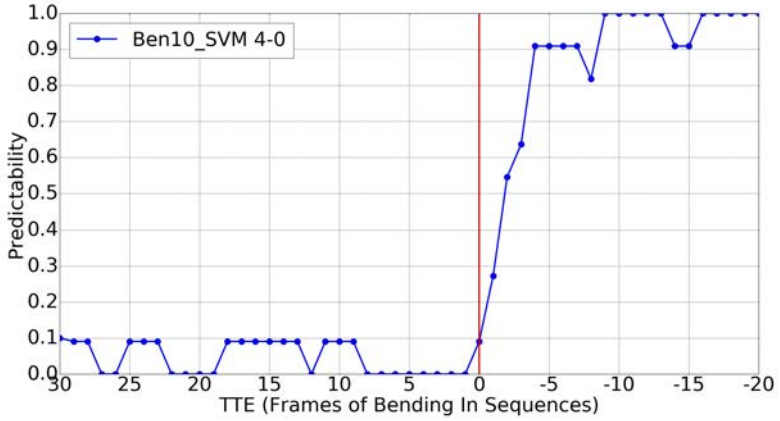
Figure 2.10 – Results for the *bending* classification task ($\mathscr{C}_b$), using GT pedestrian BBs, a time sliding window of 10, the RBF-SVM classifier and 4 – 0 as trade off for setting positive and negative frames during training. 'Ben' curve means *applied to testing bending sequences*.

2.4.5 Starting

As can be seen in Table 2.1, there are too few sequences of this type. Therefore, we have augmented the training set with frames coming from the training sequences



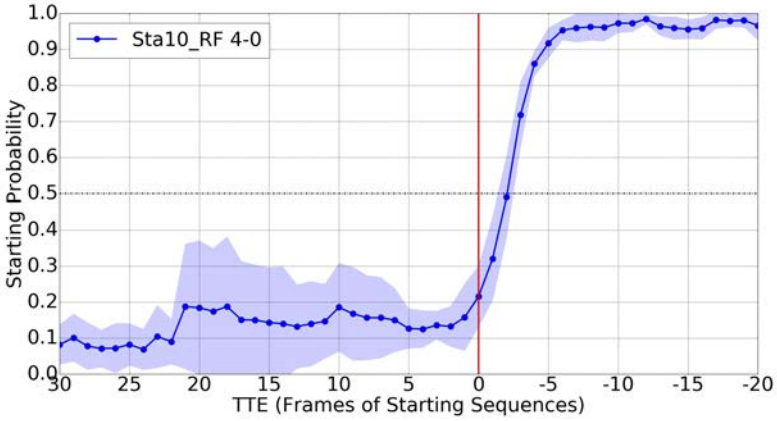
(a) Classification probability.



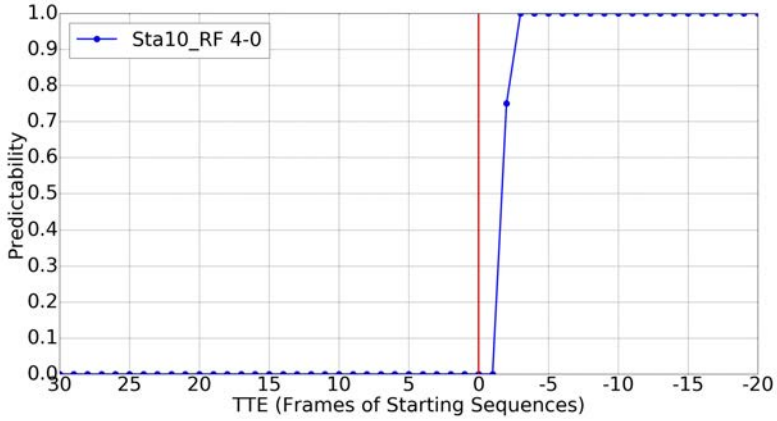
(b) Predictability for $\mathscr{C}_b > 0.16$.

Figure 2.11 – Analogous to Fig. 2.10, but using the BBs of the provided pedestrian detections.

of crossing, stopping and bending. In particular, frames from the crossing sequence are taken as positive samples of starting, as well as frames from bending sequences with $TTE < 0$ and stopping sequences with $TTE > 4$; *i.e.* all the cases when we see the pedestrians in side view walking. As negative samples we have taken frames



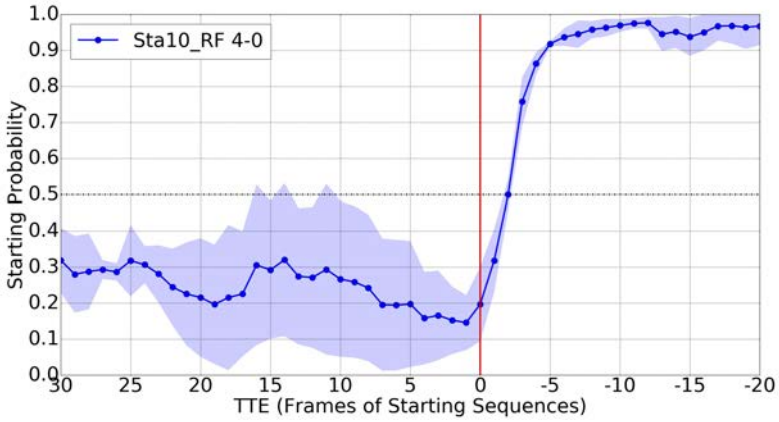
(a) Classification probability.



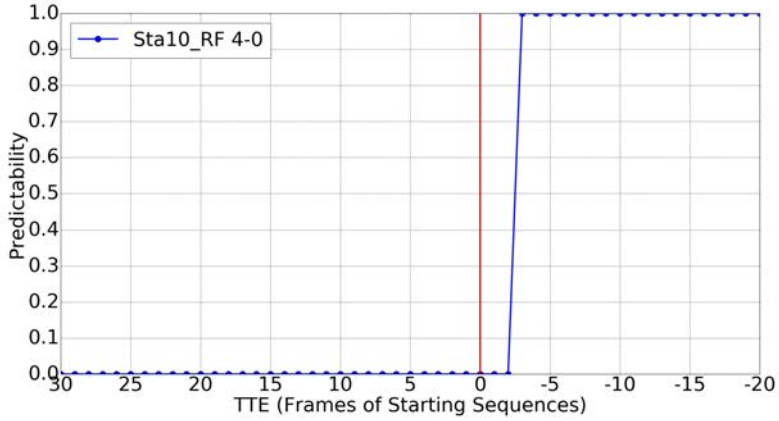
(b) Predictability for $\mathscr{C}_s > 0.50$.

Figure 2.12 – Results for the *starting* classification task ($\mathscr{C}_s$), using GT pedestrian BBs, a time sliding window of 10, the RF classifier and $4-0$ as trade off for setting positive and negative frames during training. 'Sta' curve means *applied to testing starting sequences.*

from stopping sequences with $TTE < 0$ and from bending sequences with $TTE > 4$; *i.e.* when the pedestrians are not in side view walking. At this point, we would like to



(a) Classification probability.



(b) Predictability for $\mathscr{C}_s > 0.60$.

Figure 2.13 – Analogous to Fig. 2.12, but using the BBs of the provided pedestrian detections.

comment that we tried also analogous training data augmentation for the previous classifiers ($\mathscr{C}_c, \mathscr{C}_b$), but results were more noisy, so we have not reported them here for the sake of simplicity.

As for starting, it is rather difficult to predict the action before it happens, the

aim is to understand that it is happening as soon as possible. Following [31], for training $\mathscr{C}_s$, we set the $A-B$ pair as $4-0$. In this case, we report results for $T=10$ and RF, since they are better than for RBF-SVM; but, again, values of $T \in \{1,4,7\}$ provide worse results. Fig. 2.12 shows the case for GT BBs and Fig. 2.13 for BBs from pedestrian detection. In both cases we see a predictability of over 0.8 already for TTE=3 (187ms). [31] reports TTE=4 (250ms). For $TTE>0$, Fig. 2.13a shows worse results than Fig. 2.12a due to similar reasons than in bending, *i.e.* pedestrians are further away and the detection works worse, and this may have impact on the pose estimation if the detection BBs is too noisy.

2.4.6 Crossing/not crossing

For training LSTMs, we need more data. Thus, we divide the whole sequence to crossing/not crossing (C/NC) intentions. In particular, frames from the crossing sequence are taken as C, as well as frames from bending sequences with $TTE<0$, stopping sequences with $TTE>0$ and starting sequences with $TTE<0$. The remaining sequences are taken as NC. We compare three kinds of input data for LSTMs. The first one corresponds to the coordinates of 9 keypoints, which are normalized by the ground truth BB size (GtSKLT). The second one is analogous but normalized by the shoulders and ankles (NoiseSKLT). The last input corresponds to the 396 features computed from the keypoints (RelativeSKLT). We obtain 0.96 and 0.95 as accuracy for GtSKLT/LSTM and RealtiveSKLT/LSTM, respectively. Accuracy of 0.87 is obtained by the NoiseSKLT/LSTM. However, by using the RelativeSKLT/RF, we obtained an accuracy of 0.98. This suggests that the LSTM based solution is sensible to the skeleton fitting and, in fact, is not providing better results than appending the temporal window features to a RF; while LSTMs are more costly to train than RF in terms of training data.

2.5 Summary

The state of the art on on-board detection of pedestrian intentions is not so extensive, especially compared to pedestrian detection and tracking. The proposed methods rely on dense stereo data and/or dense optical flow. In this paper we have shown how modern CNN-based off-the-shelf 2D pedestrian pose estimation methods can be used to develop a detector of pedestrian intentions from monocular images. On top of a fitted human skeleton we have defined keypoint relative features which, together with well grounded and efficient machine learning methods (SVM, RF), allowed us to address the detection of situations such as *crossing vs stopping*, *bending*, and *starting*. We showed that feature concatenation over a

time sliding window of ten frames gives rise to results that are even better than the state of the art based on processing dense stereo data. Our experiments show anticipation of 750ms regarding a pedestrian that will cross the road, 250ms after a pedestrian performs a bending action, and 187ms when a pedestrian starts to enter the road after being on a standstill state. There are still difficult cases, specially when the pedestrians are seen in back or frontal views at far distance, since then the pose estimation can fluctuate in the skeleton adjustment (confusing left and right body parts). This affects bending detection, thus, it will be one of our first addressed future works. In addition, interesting future work consists of assessing the same pedestrian intention scenarios when there are more pedestrians, eventually occluding each other; which must start by producing a proper dataset with such cases.

3 Pedestrian intention in the wild

Our recent work suggests that, thanks to nowadays powerful CNNs, image-based 2D pose estimation is a promising cue for determining pedestrian intentions such as *crossing* the road in the path of the ego-vehicle, *stopping* before entering the road, and *starting* to walk or *bending* towards the road. This statement is based on the results obtained on non-naturalistic sequences (Daimler dataset), *i.e.* in sequences choreographed specifically for performing the study. Fortunately, a new publicly available dataset (JAAD) has appeared recently to allow developing methods for detecting pedestrian intentions in naturalistic driving conditions; more specifically, for addressing the relevant question *is the pedestrian going to cross?* Accordingly, in this paper we use JAAD to assess the usefulness of 2D pose estimation for answering such a question. We combine CNN-based pedestrian detection, tracking and pose estimation to predict the crossing action from monocular images. Overall, the proposed pipeline provides new state-of-the-art results.

3.1 Introduction

Even there is still room to improve pedestrian detection and tracking, the state-of-the-art is sufficiently mature [17, 65, 66] as to allow for increasingly focusing more on higher level tasks which are crucial in terms of (assisted or automated) driving safety and comfort. In particular, knowing the intention of a pedestrian to cross the road in front of the ego-vehicle, *i.e.* before the pedestrian has actually entered the road, would allow the vehicle to warn the driver or automatically perform maneuvers which are smoother and more respectful with pedestrians; it even significantly reduces the chance of injury requiring hospitalization when a vehicle-to-pedestrian crash is not fully avoidable [41].

The idea can be illustrated with the support of Fig. 3.1. We can see two pedestrians, one apparently stopped near a curb and the other walking towards the same curb. Just looking at the location of the (yellow) bounding boxes (BBs) that frame

Figure 3.1 – Our focus: *is the pedestrians going to cross?*

these pedestrians, we would say that they are not in the path of the vehicle at the moment. However, we would like to know what is going to happen next: is the stopped pedestrian suddenly going to cross the road? is the walking pedestrian going to cross the road without stopping?; in the affirmative cases, the vehicle could start to slow down already for a safer maneuver, increasing the comfort of the passengers and the confidence of the pedestrians (especially relevant for autonomous vehicles).

Recently, we have addressed the crossing/not-crossing classification (C/NC) task by relying on image-based 2D pose estimation Chapter 2. The proposed method shows state-of-the-art results and, in contrast to previous approaches (see Sect. 3.2), it does not require information such as stereo, optical flow, or ego-motion compensation. As was common practice in the state-of-the-art literature, in Chapter 2 we used the only publicly available dataset for the C/NC task at that time, kindly released by Daimler researchers [55]. While this dataset is a good starting point to challenge different ideas, it is composed of non-naturalistic sequences, *i.e.* they show isolated pedestrians performing actions specifically choreographed for the C/NC task. Fortunately, a new dataset (*Joint Attention for Autonomous Driving*–JAAD) has been publicly released recently [46], which allows to address the C/NC task in naturalistic driving conditions. Accordingly, in this paper we present (see Sect. 3.3) a pipeline consisting on a pedestrian detector, a multi-pedestrian tracker and a 2D pedestrian pose estimator, to obtain a per-pedestrian multi-frame feature set which allows to perform the C/NC task. Detector, Tracker and Pose Estimator are based on off-the-shelf CNN modules designed for such generic tasks, which we adapt here for our C/NC task. In this way, we can perform our experiments (see Sect. 3.4) in the JAAD dataset.

Therefore, with respect to Chapter 2, we are facing a more challenging dataset for which using state-of-the-art pedestrian detection and tracking is mandatory. Note that for the dataset used in Chapter 2, it was sufficient to rely on a simple HOG/Linear-SVM pedestrian detector and no tracking since the sequences only show single pedestrians under favorable illumination conditions. Moreover, since recently CNN-based features have been used to address the C/NC task in JAAD [47], we additionally compare our pose-estimation-based features with CNN-based ones. As we will see, the former clearly outperform the latter. Even more, as additional novelty we also report time-to-event (TTE) results in JAAD, which reinforce our argument about using pose estimation for detecting the crossing intentions of pedestrians. Overall, we think we are contributing with a new state-of-the-art baseline for JAAD, which is the only publicly available dataset at the moment acquired in naturalistic driving and containing ground truth annotations for the C/NC task.

3.2 Related Work

The C/NC task was initially taken as an explicit pedestrian path prediction problem; addressed by relying on pedestrian dynamic models for estimating pedestrian future location, speed and acceleration [29, 55]. However, these models are difficult to adjust and for robustness require to rely on dense stereo data, dense optical flow and ego-motion compensation. Intuitively, methods like [29] implicitly try to predict how the silhouette of a tracked pedestrian evolves over time. In fact, [31] uses a stereo-vision system and ego-motion compensation to explicitly assess the silhouette of the pedestrians (others rely on 360° LIDAR [61]). Note that, while our method will be applied in JAAD because only relies on a monocular stream of images, these other methods cannot be applied due to the lack of stereo information and vehicle data for ego-motion compensation.

On-board head and body orientation approximations have been also proposed to estimate pedestrian intentions, both from monocular [49] and stereo [14, 56] images with ego-motion compensation. However, it is unclear how we actually can use these orientations to provide intention estimation. Moreover, the experiments reported in [56] suggest that head detection is not useful for the C/NC task.

These mentioned vision-based works relied on Daimler's dataset. By using an AlexNet-based CNN trained on JAAD, [47] verified whether full body appearance improves the results on the C/NC task compared to analyzing only the sub-window containing either the head or the lower body. Conclusions were similar, *i.e.* specifically focusing on legs or head does not seem to bring better performance.

In fact, in [54] it is concluded that *a lack of information about the pedestrian's posture and body movement results in a delayed detection of the pedestrians changing*

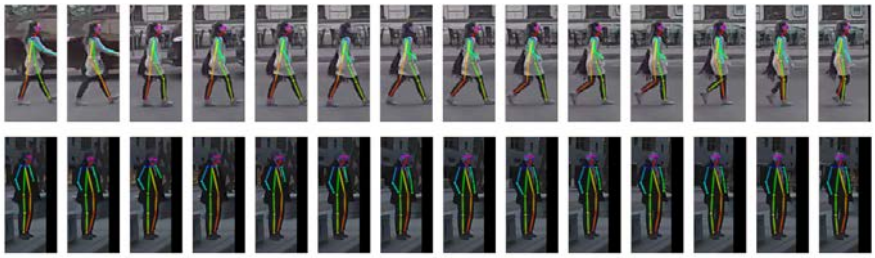


Figure 3.2 – Examples of 2D pose estimation by skeleton fitting. Top: pedestrian in side-view walking. Bottom: pedestrian standing still. From left to right we see 14 consecutive frames of two JAAD sequences, which roughly correspond to half a second.

their crossing intention. In line with this suggestion, in Chapter 2 we relied on a state-of-the-art 2D pose estimation method that operates in still images [3]. In particular, following a sliding time-window approach, accumulating estimated pedestrian skeletons over-time (see Fig. 3.2) and features on top of these skeletons (see Fig. 2.5), we obtained state-of-the-art results for the C/NC task in Daimler's dataset; which is remarkable since we only relied on a monocular stream of frames, but neither on stereo, nor on optical flow, nor on ego-motion compensation. In this paper, we augment our study to the more challenging JAAD dataset by complementing the 2D pose estimation with state-of-the-art pedestrian detection and tracking. Moreover, we compare the use of skeleton-based features with CNN-appearance-based ones as suggested in [22] for the generic task of human action recognition. We will see how the former bring more accuracy than the latter. In addition, we also report TTE results.

3.3 Method

In order to address the C/NC task we need to detect pedestrians, track them, adjust a skeleton for each one, frame by frame (see Fig. 3.2), and apply a C/NC classifier for each pedestrian by relying on features defined on top of the respective skeleton (see Fig. 2.5). Accordingly, in this section we briefly describe the components used for detection, tracking, skeleton fitting (pose estimation) and C/NC classification.

Detection For pedestrian detection we have fine-tuned a generic object detector based on the popular Faster R-CNN [51]. In particular, we have used the Tensor-

Flow publicly available implementation described in [5], based on a VGG16 CNN architecture. During the training stage of the C/NC classification pipeline, we have fine-tuned the model with JAAD training images.

Tracking Pedestrian tracking is addressed as a multiple object tracking-by-detection paradigm. A state-of-the-art tracker addressing this paradigm can be found in [65], which has associated publicly available code that we have used out-of-the-shelf. This tracker uses the following *state* for a pedestrian detection: $(u, v, \lambda, h, \dot{x}, \dot{y}, \dot{\lambda}, \dot{h})$; where (u, v) represents the central pixel of the BB, λ is its aspect ratio, h its height, while $\dot{x}$, $\dot{y}$, $\dot{\lambda}$, and $\dot{h}$ are the respective velocities. These state variables are updated according to Kalman filtering. For performing data association, it is used a cosine distance on top of CNN features (trained on a large-scale person re-identification dataset [67]) which scores the degree of visual similarity between BB detections and predictions. A detection which does not have a high matching score with some prediction is pre-tracked; if the lack of matching holds during several consecutive frames (for JAAD we set 3 frames, *i.e.* 0.1 seconds), the track is consolidated as corresponding to a new pedestrian. Predictions which do not have a high matching score with a new detection during several frames (for JAAD we set 30 frames, *i.e.* 1 second) are considered as disappeared pedestrians (ended tracks). Note that this tracking process is purely image-based, no ego-motion compensation is required.

Skeleton fitting (pose estimation) Given the good results obtained in Chapter 2, we apply the CNN-based pose estimation method proposed in [3], which has publicly available code. This method can operate in still monocular images and has been trained on the *Microsoft COCO 2016 keypoints dataset* [39]. It is supposed to perform both pedestrian detection and pose estimation. However, in our initial experiments with JAAD dataset, detection itself was not as good as Faster R-CNN. We think this is because, while we fine-tuned the later with JAAD images, we did not do the same for the pose estimation method since it would require annotations at pedestrian body level. Thus, what we do is to run the pose estimation only within the BBs predicted by the tracking system, obtaining in that way the desired skeletons (Fig. 3.2).

C/NC classification In Chapter 2 we extracted features from the fitted skeleton and use them as input to a classifier (SVM/Random Forest). Fig. 2.5 shows that the fitted skeleton is based on 18 keypoints. We use the most stable 9 keypoints highlighted with a star, which correspond to the legs and the shoulders. These are highly relevant keypoints since the legs execute continue/start walking or stopping actions; while keypoints from shoulders and legs inform about global body

orientation. From the selected keypoints we compute features. First, we perform a normalization of keypoint coordinates according to a factor h proportional to the pedestrian height (Fig. 2.5). Then, different features (conveying redundant information) are computed by considering distances and relative angles between pairs of keypoints, as well as triangle angles induced by triplets of keypoints. In total we obtain 396 features. Since we concatenate the features collected during the last T frames, our feature vector has dimension $396T$. In addition, for comparison purposes, as in the general action recognition literature [22], we also test the $fc6$ features provided by the Faster R-CNN at each pedestrian BB; a $4096T$ dimensional vector. Finally, since Random Forest (RF) directly provides a probability measure for a meaningful thresholding, we use it for performing the C/NC classification based on the selected features (skeleton or $fc6$ based ones).

3.4 Experiments

3.4.1 Dataset

First publicly available dataset for research on detecting pedestrian intentions is from Daimler [55]. It contains 68 short sequences (9,135 frames in total) acquired in non naturalistic conditions and shows a single pedestrian per video, where the pedestrian is forced to perform pre-determined actions. More recently, it has been publicly released the Joint Attention for Autonomous Driving (JAAD) dataset [46], acquired in naturalistic conditions and annotated for detecting C/NC actions. It contains 346 videos (most of them 5-10 seconds long) recorded on-board with a monocular system, running at 30 FPS with a resolution of 1920×1080 pixels. Videos include both North America and Eastern Europe scenes. Overall, JAAD includes $\approx$ 88,000 frames with 2,624 unique pedestrians labeled with $\approx$ 390,000 BBs. Moreover, occlusion tags are provided for each BB. Where $\approx$ 72,000 (18%) BBs are tagged as partially occluded and $\approx$ 46,000 (11%) as heavily occluded. In addition, although we are not using it in this paper, JAAD contains also context information (traffic signs, street width, etc.) that we may use in further studies to complement purely pedestrian-based information.

3.4.2 Evaluation protocol

In [47], JAAD was used for assessing a proposed C/NC method. However, it is not explained how the JAAD data was divided into training and testing, and the corresponding code is not available. Therefore, here we have followed a protocol that we think is reasonable and can be reproduced. First of all, We take the first

250 videos of JAAD for training and the rest for testing. Moreover, pedestrians are labelled with many different actions which we have mapped to C/NC as follows. We term as C to the crossing labels of JAAD, as well as the labels in {clear-path, moving-fast, moving-slow, slow-down, speed-up} assigned to a pedestrian with lateral motion direction; the rest are denoted as NC.

Training

In order to fine-tune the Faster R-CNN we consider all the training frames and basically follow the same settings than in [5], but using {8, 16, 32, 64} as anchors and 2.5 as BB aspect ratio (*i.e.* pedestrian oriented). For fine-tuning we perform 110,000 iterations (remind that an iteration consists of a batch of 256 regions from the same image, and that input images are vertically mirrored to double the number of training samples). Regarding learning rate, we start with 0.001 and decrease the value to 0.0001 after 80,000 iterations.

In order to train the C/NC classifier we needed to rely on well seen pedestrians as well as balancing the number of samples of the C and NC classes. For achieving this goal, we only consider pedestrian training samples with a minimum BB width of 60 pixels and no occlusion. Moreover, for a tracked pedestrian, these conditions must hold over more than T frames, since we need to concatenate last T frames for the C/NC classification. Thus, from tracks longer than T frames we can obtain different training samples by applying a temporal sliding window of size T.

For each tracked pedestrian, the C/NC label assigned to a generated sequence of length T corresponds to the label in the most recent frame (*i.e.* the frame in which the C/NC decision must be taken). We set T=14 for JAAD (*i.e.*, following Chapter 2, a value roughly below 0.5 seconds). Note that, since we are in training time, here we are referring to the ground truth tracks provided in JAAD. For completeness, we also test the case T=1; meaning that we only train with the last frame of the same sequences used for the T=14 case. Overall, there are 8,677 sequences of length T=14 and NC label, while there are 36,253 with C label; thus, in the latter case we randomly take only 8,677 among those 36,253 possible. Accordingly, we fit the pose estimation-based skeleton and compute the C/NC features (Fig. 2.5) for 8,677 C and 8,677 NC samples (in a set of experiments for T=14, in another for T=1). These features are then used as input for the scikit-learn [45] function GridSearchCV; which is parameterized for training a Random Forest (RF) classifier using 5-fold cross-validation with the number of trees running on $\{100, 200, 300, 400, 500\}$ and maximum depth running on $\{7, 15, 21, 30\}$. The optimum RF in terms of accuracy corresponds to 400 trees and a maximum depth of 15, but we noted that all configurations provided very similar accuracy.

In order to compare skeleton-based features with CNN-based ones, we apply

the following procedure. For all training images we run the VGG16 obtained during Faster R-CNN fine-tuning. Then, for the same tracks mentioned before, we replace the skeleton-based features by the fc6 layer features inside the tracked pedestrian BBs. Note that (Sect. 3.3) we have $396T$ skeleton-based features and $4096T$ fc6-based ones for each sample reaching RF training. In terms of RF parameter optimization (number of trees and maximum depth), CNN-based features reported similar accuracy as was the case for skeleton-based ones. Therefore, we set the same parameters, *i.e.* 400 trees and a maximum depth of 15. For the sake of completeness, we also combine skeleton and CNN-based features using the same RF parameters.

Testing

In [47] evaluations are single frame based (T=1 in our notation) and only pedestrians with an action label are considered (those mapped to C/NC) here. When designing our experiments, we have seen that not all pedestrians of JAAD are annotated with a BB. Therefore, when we run the detection and tracking modules, we are detecting and tracking some pedestrians which do not have the required ground truth information (BB, etc.). So, in order to follow a similar approach to [47], we do not consider these cases for quantitative evaluation. However, they are present in the qualitative evaluation (*e.g.* see the videos provided as supplementary material). Overall, we ensure that T=1 and T=14 experiments are applied at the same tracked pedestrians at the same frames, so we perform a fair comparison.

When detecting pedestrians with Faster R-CNN we use the default threshold 5% and overlapping of 30% for non-maximum suppression. For starting a new track, a pedestrian must be detected in 3 consecutive frames; while for ending a track there must be no new matched observations (detections) during 30 frames. For pose estimation (skeleton fitting) we use 3 scales; in particular, $\{1, (1-0.15), (1-0.15*2)\}$. For the C/NC classifier we use 0.5 as classification threshold.

We assess accuracy according to the widespread definition $Acc = (TP+TN)/(P+N)$, where P stands for total positives (here "C"), N are the total negatives (here "NC"), and TP and TN the rightly classified positives and negatives (C and NC right classifications). According to the testing protocol we have defined, we found $P = 17045$ and $N = 5161$, therefore, Acc could be bias towards "C" results. In order to avoid this, we select $P = N$ cases randomly. Thus, Acc will be based on 10,322 testing decisions.

In addition, similar to Chapter 2, we are interested in providing time-to-event (TTE, (see Appendix Sect. A.2 for more details)) results for the critical case of crossing (C). However, JAAD is not annotated for this. Then, we added the TTE information to 9 sequences we could describe as *keep walking to cross*, and 14

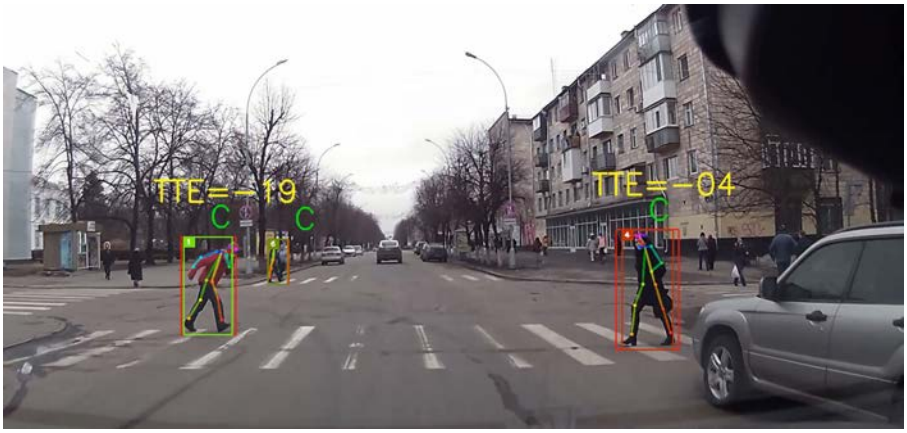


Figure 3.3 – Results of C/NC classification. The ground truth label is indicated with a "C" or a "NC"; when written in green color, it means that the prediction agrees with the ground truth, otherwise it would be written in red. Pedestrians are framed with two BBs: detection and tracling ones, the latter with the corresponding track ID. The estimated pedestrian skeleton is also shown. When annotated, time-to-event (TTE) is also shown in frame units. Negative TTE values mean that the even happened before this frame, while positive values indicate that it will happen after.

more sequences we could describe as *start walking to cross*. TTE = 0 is when the event of interest happens. Here we consider separately (a) pedestrians walking towards a curbside without stopping, just entering the road; and (b) pedestrians standing close to the curbside that start to walk entering the road. Positive TTE values correspond to frames before the event, negative values to frames after the event. Fig. 3.3 shows a result example where we can see TTE values for different pedestrians that are correctly classified as crossing (the supplementary videos have more examples). With TTE we provide two different plots, *intention probability vs* TTE, and *predictability vs* TTE. With the former we can see how many frames we can anticipate the pedestrian action. Since there are several testing sequences per intention, mean and standard deviation are plotted. *Predictability* plots show a normalized measurement of how feasible is to detect the action under consideration for each TTE value. Predictability zero indicates that we cannot detect the action, while predictability one means that we can.

Table 3.1 – Classification accuracy (Acc) in JAAD. SKLT stands for the use of our skeleton-based features, while CNN (fc6) are the features we take from a VGG16 fine-tuned in JAAD (see main text). We have included here the results reported in [47], where CNN features are based on a non-fine-tuned AlexNet and Context refer to features of the environment, not of the pedestrian itself. Moreover, results for 20% and 30% noise in the keypoints is also reported for the SKLT case (se main text for details).

Method	**T**	**features**	**Acc**	**Acc** 20%	**Acc** 30%
[47]	1	CNN	0.39		
[47]	1	$CNN\&Context$	0.63		
$Ours$	1	$CNN(fc6)$	0.68		
$Ours$	1	$SKLT$	0.80	0.77	0.73
$Ours$	1	$CNN(fc6)+SKLT$	0.81		
$Ours$	14	$CNN(fc6)$	0.70		
$Ours$	14	$SKLT$	**0.88**	0.86	0.83
$Ours$	14	$CNN(fc6)+SKLT$	0.87		

Results

Table 3.1 reports the accuracy results. In the sake of completeness, we have included those reported in [47]; however, our results are not directly comparable since it is unclear which frames where used for training and which ones for testing. The paper mentions that heavily occluded pedestrians are not considered for testing. In our experiments we do not exclude pedestrians due to occlusion. Moreover, we also report TTE information. However, we still found interesting to include the results in [47] since the paper is based on CNN features and T=1. In particular, the authors train a walking/standing classifier and another looking/not-looking (pedestrian-to-car) classifier, both classifiers are based on a modified AlexNet CNN. Actually, the classification score of these classifiers are not used for final C/NC decision. Instead, the fc8 layer of both are used as features to perform a final C/NC based on a Linear-SVM adjusted in such a CNN-based feature space. It is also proposed to add contextual information captured by a place-classification style AlexNet.

From Table 3.1, we can see that for a fixed T the features based on the skeleton of the pedestrian (SKLT) outperform those based on CNN fc6 layer. Combining

SKLT and fc6 does not significantly improves accuracy of SKLT. We can see also that T=14 outperforms T=1, showing the convenience of integrating different frames. From Fig. 3.4 to Fig. 3.7, we can see that the system is stable at predicting that a walking pedestrians will keep moving from a sidewalk and eventually crossing the curbside appearing in front of the vehicle. We can see also that we can predict (predictability>0.8) that a standing pedestrian will cross the curbside around 8 frames after he/she starts to move, which in JAAD is around 250ms.

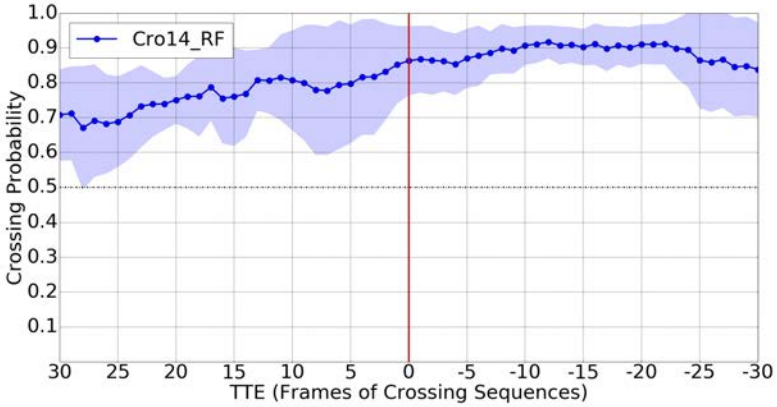


Figure 3.4 – *Keep walking to cross*, T=14. Blue curve: mean over sequences; blue area: standard deviation.

Looking in more detail to the results, we find situations that need to be taken into account as future work. For instance, in Fig. 3.8 there is a "C" accounted as error (red). Indeed, the pedestrian is crossing the road, but not the one intersecting the path of the ego-vehicle. So in the evaluation it should be probably accounted as right. On the contrary, in Fig. 3.9 the system classifiers as "NC" a pedestrian which is not crossing the road, but in fact is walking along the road, in front of the car. Now this situation is accounted as right, but probably should be accounted as wrong. On the other hand, in this case we can just use location-based reasoning to know that the pedestrian is in a dangerous place, it is not a problem of predicting the action anymore (as the C/NC case). It is worth also mentioning that we have observed that walking in parallel to the car motion direction, tends to be properly classified as NC; however, more annotations are required to provide a reasonable quantitative analysis. Check our demo for more information (https://youtu.be/we4weU0NSGA).

In order to evaluate the robustness of the method, we ran an equivalent set of

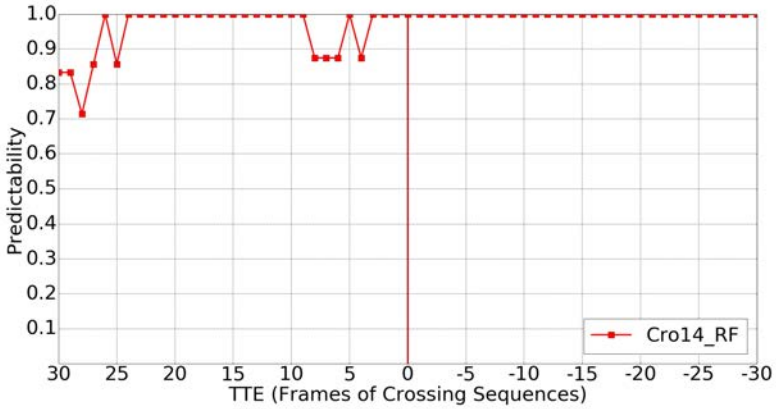


Figure 3.5 – *Keep walking to cross*, T=14, prob. thr. = 0.5.

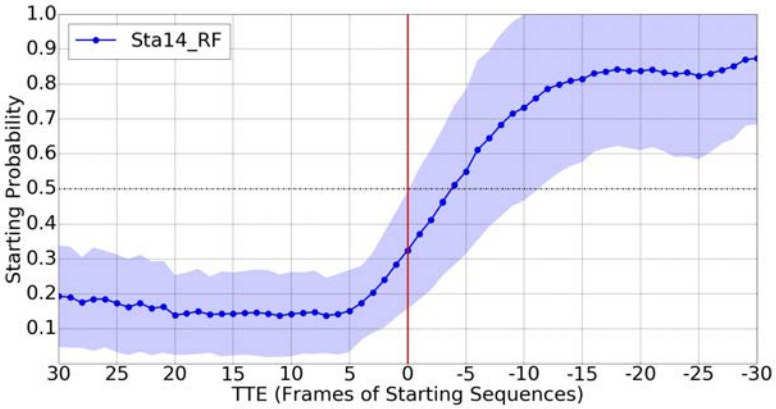


Figure 3.6 – *Start crossing*, T=14.

experiments for the SKLT case, where we added random noise to the keypoints of the fitted skeleton in testing time. In particular,independently for each coordinate of each keypoint, we added Gaussian noise with zero mean and standard deviation s which, following [28], is set as a percentage over the distance to the closest keypoint. This is shown in Table 3.1 for percentages of 20% and 30%. As expected, accuracy

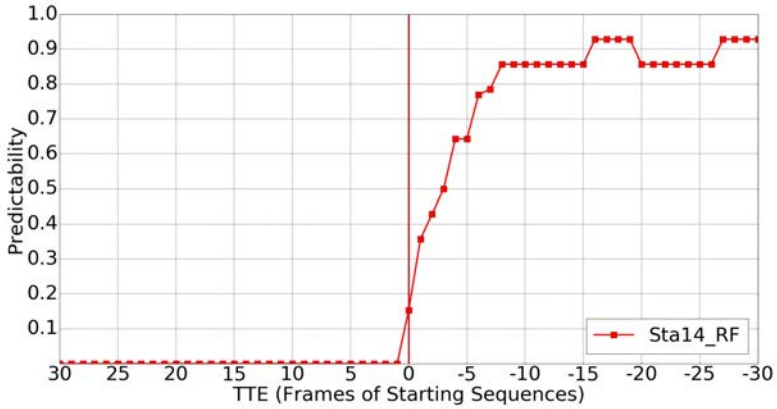


Figure 3.7 – *Start crossing*, T=14, prob. thr. = 0.5.

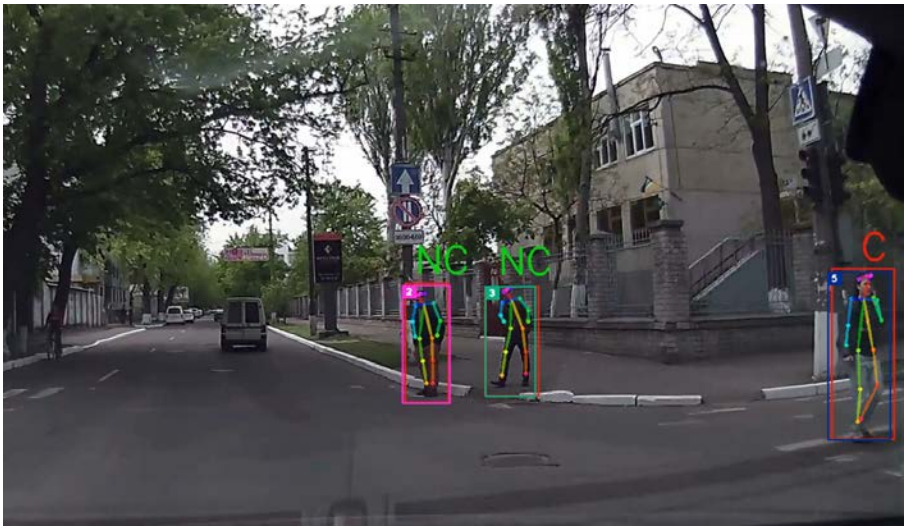


Figure 3.8 – Results of C/NC classification

decreases a few for 20% and more for 30%, being T=14 is more robust to noise than T=1 (see Appendix Sect. A.3 for more details).

Finally, we assess the most important features for the RF classifier. In Fig. 3.10

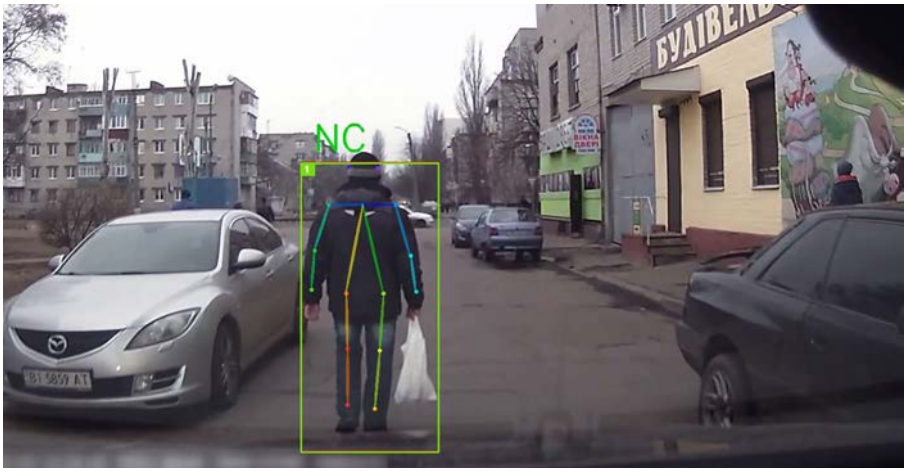


Figure 3.9 – Results of C/NC classification

we assign an ID to the keypoints of a fitted skeleton either used for pedestrian or cyclist intention recognition. The visualized naming scheme defines angles ($\Theta(\cdot,\cdot),\Theta(\cdot,\cdot,\cdot)$) and lengths ($L(\cdot,\cdot),L_x(\cdot,\cdot),L_y(\cdot,\cdot)$). Since we have evaluated both T=1 and T=14, in the latter case we also add a super-index to indicate from which relative frame index (*i.e.* in $\{1,..,14\}$) comes the feature. Tables 3.2-3.3 show the top-25 more important features for T=1 (top ~ 6%) and T=14 (top ~ 0.5%), respectively. We see how all are based on 3-keypoint angles, mostly connecting either shoulder and legs, or shoulder and waist; thus, capturing global pose. For T=14, only one feature appears after frame 9 (~ 300ms); thus, favoring intention prediction in a short time.

Table 3.2 – For T=1, top-25 most relevant pedestrian skeleton-based features from left-to-right and top-to-bottom.

$\Theta(4,12,3)$	$\Theta(4,11,3)$	$\Theta(8,4,9)$	$\Theta(4,8,3)$	$\Theta(3,10,5)$
$\Theta(8,12,10)$	$\Theta(12,11,13)$	$\Theta(3,12,5)$	$\Theta(4,13,3)$	$\Theta(3,8,5)$
$\Theta(4,10,3)$	$\Theta(8,5,9)$	$\Theta(8,3,9)$	$\Theta(3,11,5)$	$\Theta(12,10,13)$
$\Theta(4,9,3)$	$\Theta(10,8,12)$	$\Theta(3,9,5)$	$\Theta(8,10,12)$	$\Theta(3,8,9)$
$\Theta(3,9,8)$	$\Theta(3,13,5)$	$\Theta(9,13,11)$	$\Theta(12,9,13)$	$\Theta(12,8,13)$

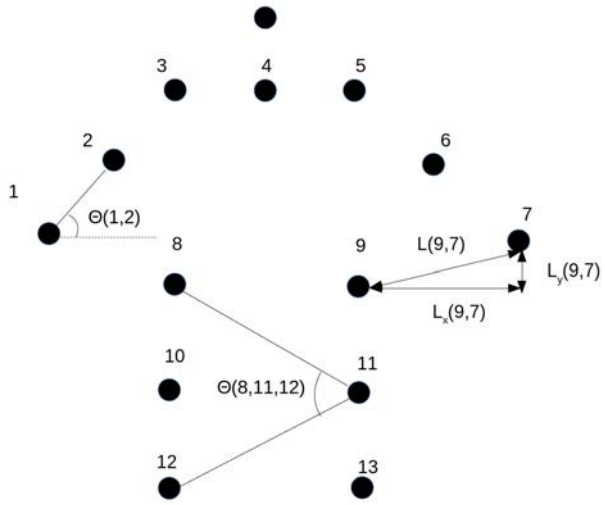


Figure 3.10 – Naming scheme for skeleton-based features.

Table 3.3 – For T=14, top-25 most relevant pedestrian skeleton-based features from left-to-right and top-to-bottom.

$\Theta^{7}(4,9,3)$	$\Theta^{3}(4,12,3)$	$\Theta^{8}(4,13,3)$	$\Theta^{9}(4,12,3)$	$\Theta^{2}(4,12,3)$
$\Theta^{6}(4,9,3)$	$\Theta^{12}(4,8,3)$	$\Theta^{9}(4,10,3)$	$\Theta^{4}(4,10,3)$	$\Theta^{1}(4,9,3)$
$\Theta^{4}(4,8,3)$	$\Theta^{8}(4,12,3)$	$\Theta^{6}(4,8,3)$	$\Theta^{8}(4,10,3)$	$\Theta^{6}(4,13,3)$
$\Theta^{6}(8,10,9)$	$\Theta^{1}(4,12,3)$	$\Theta^{4}(3,8,5)$	$\Theta^{2}(4,8,3)$	$\Theta^{3}(4,8,3)$
$\Theta^{7}(4,8,3)$	$\Theta^{4}(4,9,3)$	$\Theta^{7}(4,10,3)$	$\Theta^{5}(4,11,3)$	$\Theta^{8}(4,9,3)$

3.5 Summary

In this paper we have evaluated a fully vision-based pipeline (detection, tracking and pose estimation) to address the pedestrian crossing/not-crossing problem, in naturalistic driving conditions (JAAD dataset). We show that integrating pedestrian pose based features along time, gives rise to a powerful crossing/not-crossing classifier. As to the best of our knowledge, at the moment this paper establishes the state-of-the-art results for the JAAD dataset.

4 Cyclist arm signal recognition

Anticipating the intentions of cyclists can be critical for performing safe and comfortable driving maneuvers. This is the case for human driving and, therefore, should be taken into account by systems providing any level of driving assistance, *i.e.* from advanced driver assistant systems (ADAS) to fully autonomous vehicles (AVs). In this chapter, we show how the latest advances on monocular vision-based human pose estimation, *i.e.* those relying on deep Convolutional Neural Networks (CNNs), enable to recognize the intentions of such cyclists. We assume that cyclists follow the established traffic codes to indicate future left/right turns and stop maneuvers with arm signals. In this chapter, we show how the methodology applied in recognizing pedestrian' intentions can be used for recognizing cyclists' intentions. For cyclists intention recognition, we did not found an dataset, therefore, we created our own one by acquiring and annotating corresponding video-sequences which we aim to share with the research community. Overall, the proposed pipeline provides new state-of-the-art results on the intention recognition of cyclists.

4.1 Introduction

Recognizing the motion intentions of cyclists is highly relevant since many times the ego-vehicle will need to overtake them. While we cannot assume that pedestrians will explicitly indicate their intentions, in the case of cyclists we can exploit traffic rules. In particular, as illustrated in Fig. 1.4, cyclists must indicate future left/right turns and stop maneuvers with arm signals.

In this chapter, we explore the idea of using 2D pose estimation from monocular images as core information to recognize the intentions of cyclists. The main problem is the lack of publicly available video datasets designed to assess arm signal recognition. Therefore, as first step, we acquired our own dataset using a consumer-graded camera (as in JAAD) and annotated the arm signals performed by cyclists. In total, we have annotated 219 arm signal actions on videos of approximately

10 seconds each, containing one or two actions per video. We also annotated 10 additional arm signals on Youtube videos. As additional contribution of this work, we aim at publicly releasing the annotated cyclist dataset. Using these datasets, we will show how the same pipeline for recognizing pedestrian intentions can work for cyclist intention recognition too.

The rest of the chapter is organized as follows. In Sect. 4.2, we summarize most related work. In Sect. 4.3, we present the proposed pipeline to recognize cyclist intentions. In Sect. 4.4, we detail the devised experiments, the obtained results, and the conclusions derived from them. We present also the above mentioned dataset for cyclist intention recognition. Finally, Sect. 4.5 summarizes the presented work and its possible continuations.

4.2 Related Work

Compared to pedestrian intention recognition, recognizing cyclist arm signals has received less attention so far. One core reason may be the lack of publicly available datasets for addressing this functionality. After [37, 38], it was publicly released one of the largest datasets focusing on cyclists, termed as Tsinghua-Daimler Cyclist Benchmark dataset (TDCB); however, acquired data and annotations are intended to support detection and orientation estimation tasks, but not cyclist arm signal recognition. In [1], the ground truth of TDCB was extended with wheel annotation for the case of bikes in side view; however, this is intended to support cyclist detection. Therefore, in this paper, we introduce our Cyclist Arm Signal Recognition dataset (CASR) containing 40,218 frames organized as short videos containing a cyclist arm signal each, in total, 219 annotated actions. For assessing generalization we also annotated 10 additional actions from Youtube, corresponding to 1,636 frames in total. CASR will be publicly released.

Using a stereo camera setup, in [15, 32] it is detected whether the left arm of a cyclist observed from the back is up or down, which is used as a context cue within a path prediction module. However, an isolated accuracy analysis of such up/down arm classification is not performed. In order to perform such a classification, the disparity map computed from stereo image pairs is used to produce a binary mask of each detected cyclist, and template matching is applied to determine if the mask correlates with a left arm up or down. In particular, the scores of matching against multiple templates, the disparity values, and the image intensities, are used as core information to build a Naive Bayesian Classifier with uniform prior, which is responsible for the desired up/down arm classification. In this paper, we do not assume stereo data and we not only account for the cyclist signal to turn left, but also to turn right (two types as shown in Fig. 1.4) and stopping. Moreover, we apply

Figure 4.1 – System modules. Framed in black background those which are specific for intention recognition. Note how we are using the same pipeline than for pedestrian intention recognition (see Fig. 2.2)

exactly the same procedure for pedestrian intention recognition and for cyclist arm signal recognition. On the other hand, the classification output of our method could be also integrated as a cue for the path prediction module of [15, 32].

Finally, for the sake of completeness, we would like to mention the Waymo's US Patent [33]. In the described approach arm signal recognition is based on LiDAR data, while we rely only on monocular vision. [33] does not report results on any specific dataset; however, we think this LiDAR-based approach and ours can be complementary in both early and late fusion recognition settings.

4.3 Method

The cyclist arm signal classification task is analogous to the pedestrian intention recognition Chapter 3, since it requires detection, tracking, and skeleton fitting (pose estimation), as preliminary steps for performing the classification. Fig. 4.1 depicts the modules of the system in a nutshell. We consider that detection and tracking modules would eventually be part of ADAS and AVs, *i.e.* these are not tasks forced by the intention recognition method proposed in this chapter. Skeleton fitting and intention classification (feature extraction and classifier application) modules, are specific for the tasks considered in this chapter. In the rest of the section we describe these components, *i.e.* detection, skeleton fitting, and intention classification.

Detection. In our proposal for intention recognition, one can always take a state-of-the-art cyclists (object) detector as long as it only requires a single RGB image as input, and returns a set of bounding boxes (BBs), each one framing a cyclist (we mean that the BB can frame only the rider or both the rider and the bike). Due to high accuracy as object detector reported recently, here we have considered Mask R-CNN [24], which can be found in the Detectron [21] frameworks.

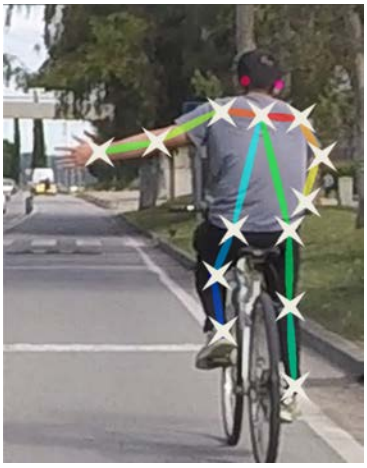


Figure 4.2 – Keypoints used for detecting the intentions of cyclists. 13 keypoints are used to extract 1170 features.

Skeleton fitting (pose estimation). Given the good results obtained in Chapter 2, we apply the CNN-based pose estimation method proposed in [3], which has publicly available code. This method can operate in still monocular images and has been trained on the *Microsoft COCO 2016 keypoints dataset* [39]. Thus, a priori it could be effective to fit the pose of cyclists. We run the pose estimation module only inside the tracked BBs, obtaining in that way the desired skeletons (see Fig. 4.2). Note that this is the same method used for detecting pedestrian intentions, which has given us excellent results.

Intention classification. We follow the similar method in Chapter 3. The pose of a cyclist is rather different than the pose of a pedestrian walking or standing. However, we hypothesize that, for performing arm signal classification, we can rely on the same keypoints than for pedestrian intention classification plus the two additional keypoints from each arm (elbow and wrist, see Fig. 4.2). Therefore, for each cyclist we use 13 keypoints, which turns out in 1170 features per frame, thus, $1170T$ for a temporal (sliding) window of T frames.

Table 4.1 – Cyclist arm signals in CASR and some YT videos.

	Turn Left	**Turn Right**	**Stop**
Cyclist 1	24	38	34
Cyclist 2	16	24	30
Cyclist 3	6	12	26
Cyclist 4	2	3	4
Total CASR	**48**	**77**	**94**
Total YouTube	**6**	**4**	**0**

4.4 Experiments

4.4.1 Dataset

As for pedestrian detection, there are large datasets for cyclist detection such as the already mentioned TDCB [15, 37]. However, it does not include samples with annotations to assess arm signal recognition. Therefore, in this paper, we introduce our Cyclist Arm Signal Recognition dataset (CASR), consisting of 40,218 frames. Moreover, for assessing generalization we also annotated additional videos from YouTube, consisting of 1,626 frames. This data will be publicly released. For CASR we followed a similar approach than JAAD authors. In particular, we attached a GoPro camera to the windshield of a car, forward facing the road ahead. We set the acquisition to RGB images at 30 fps and an 1920×1080 resolution.

We asked four persons to drive their bikes inside our university campus, and they were instructed to ride around as they wish but using arm signals when required. Sometimes they wear helmet, sometimes not. Sometimes they carry a bag in their back, sometimes not. YouTube videos are also based on a dash cam facing the road. Table 4.1 summarizes the number of actions (cyclist arm signals) that we have annotated. Note that CASR includes 219 annotated actions, and YouTube 10. Actions have been organized as short videos of around 10 seconds with a single cyclist, where the frame starting an action and the frame ending this action are annotated. The videos of CASR mostly show one action and sometimes two actions because they were indicated in a continuous way by the cyclists, and in this case we did not split the video. In addition to the frame level action annotations, we have annotated the 2D BBs framing the cyclists too. Moreover, the videos where selected so that in most of them no pedestrians are included; thus, ready to focus on cyclist arm signal recognition. In some cases, however, there can be some pedestrians but we do not annotate his/her BB so that they are ignored during training and testing. Overall, CASR's content is analogous to the first deployed dataset for pedestrian

intention recognition [55], but including much more annotated frames (68 actions within 9,135 in [55], 219 actions within 40,218 frames here).

Another important point to clarify is that action annotations are vehicle-centric, instead of cyclist-centric. When the ego-vehicle follows the cyclist, they are the same. However, when the cyclist and the ego-vehicle move in opposite directions, we annotated as left-turn what for the cyclist is an indication of right-turn, and vice versa. The reason, is that for the vehicle what matters is the direction that the cyclist is going to take as seen in the image to be processed. Figure 4.3 clarifies the idea.

4.4.2 Evaluation protocol

In CASR we recorded four cyclists. Accordingly, in order to evaluate our arm signal classifier, we divide their videos in training, validation, and testing sets. We use the videos of two cyclists for training, the videos of the other two cyclists are used for validation (training time) and testing, respectively. By varying the role of the cyclists, we can perform 12 training-validation-testing runs. Moreover, for each trained classifier, we test on the annotated YouTube videos too. We report individual metrics for each trained classifier, as well as averaged metrics. Since we aim at performing per-frame arm signal classification, we use the F1 and Accuracy standard metrics by counting classification errors and successes in each tested frame. For each training-validation run, we performed RF hyper-parameter search for the number of trees and the maximum depth allowed before performing the corresponding testing. For the former, we validated over the set $\{50, 100, 200, 300, 400, 500\}$, and for the later over the set $\{7, 10, 13, 16, 19, 22, 25, 28, 31, 34\}$. In this case, we also consider T=1 and T=14 as temporal sliding window sizes.

Since CASR and the annotated YouTube videos contain a single cyclist and no pedestrians, detecting the cyclist is sufficient to perform our evaluation, *i.e.* we do not need to run an additional tracking step. Moreover, since the human pose estimation method that we use [3] performs the double task of searching the human and fitting its skeleton in a given 2D BB, we first relied on the fine-tuned Faster R-CNN detector described in Chapter 3 for providing such BBs. However, it turned out that usually was leaving the arms of the cyclists out of the BBs. Therefore, since our focus is not on detection itself we changed the method to Mask R-CNN [24], in this case we did not fine tuned the detector to CASR since for this, object level silhouette ground truth is required. In practice, Mask R-CNN was providing accurate detections in CASR, and only in a few frames the corresponding detections were missing. In these frames, since we are not running a tracker, we just took a noisy version of the ground truth BB as detection. In particular, we added uniform noise to the BB's corner coordinates, being the amount of noise proportional to the size of the BB (we added independent Gaussian noise to each corner coordinates, with

(a) Annotated as turning right

(b) Annotated as turning left

(c) Annotated as stopping

Figure 4.3 – Annotation of cyclist arm signals. We have followed a vehicle-centric criterion for left/right annotation.

zero mean and standard deviation set to 10% of the BB height for the y coordinates, and analogously for the BB width and x coordinates). With this protocol we focus on the cyclist arm signal recognition itself, simplifying the detection-tracking pipeline given the design of CASR and the YouTube annotated videos. Moreover, Chapter 2 shows that the conclusions on pedestrian intention recognition that we draw in Chapter 2 using a dataset designed for such a task, also hold in naturalistic traffic

conditions. Thus, we expect the same for cyclist arm signal recognition.

4.4.3 Results

Tables 4.2 and 4.3, show the quantitative results for $T = 1$ (single frame), and $T = 14$ (roughly half a second), respectively. These confirm the effectiveness of the proposed method with relatively high accuracy values, which are quite stable (very low standard deviation). Testing in the YouTube videos is more challenging, but still the accuracy is remarkable since we trained the model on CASR cyclists. In both cases, aggregating temporal information does not help significantly to boost performance; which can be expected since it is already possible to understand what the intentions of the cyclists are looking at a single frame. Still, analyzing more frames can help to stabilize the classification output as we are going to see. In order to evaluate the robustness of the method, we run an equivalent set of experiments where we added random noise to the keypoints of the fitted skeleton in testing time. In particular,independently for each coordinate of each keypoint, we added Gaussian noise with zero mean and standard deviation s which, following [28], is set as a percentage over the distance to the closest keypoint. Table 4.4 compares overall accuracies for noises of 20% and 30%, both for T=1 and T=14. As we can see, only 30% causes an appreciable drop on performance for T=1, which is avoided up to a large extent thanks to the multiframe setting, *i.e.* T=14 (see Appendix Sect. A.3 for more details).

Figures 4.4-4.7 present examples of correct qualitative results for T=14. The blue BB is the detection. Note how the predictions works for forward and backward faced pedestrians, even if they carry a bag in the back, and at different distances (bigger characters correspond to further away detections). On the other hand, Fig. 4.8 shows some isolated frames with wrong predictions for CASR, also for T=14. From left to right, the two first cases correspond to oncoming cyclists indicating the intention of stopping and turning to their right (left in vehicle-centric coordinates), but no action is recognized because the detection BBs left the arms out affecting the fitting of the skeleton. This could be solved by systematically augmenting the BB size which is taken as area of interest by the skeleton fitting procedure, at testing time. In the third case the system confuses a future turn left with a stop indication, however, this is the case only at the starting of the action because it is not really clear what the cyclist is going to indicate. The next frames make it clear so that the system actually predicts the proper maneuver. In the fourth case, the system recognises that the cyclist is indicating an action, however, a stop sign is confused with a turn left, which happens because of the relatively straight position of the arm. In this case, we are able to understand the stop indication because of the hand, which is not involved in the analysis of the image. Therefore, this may suggest that a via to

Table 4.2 – Classification accuracy (**Acc**) and **F1** score for $T = 1$, both ranging from 0 to 1. **Train-Val-Test** refer to the cyclist ID of CASR used for training, validation, and testing, respectively. This turns in 12 runs. For each run, we also report generalization results on the annotated YouTube videos (**Acc-YT**, **F1-YT**). The average and standard deviation of each metric is also reported.

Train−Val−Test	**Acc**	**F1**	**Acc-YT**	**F1-YT**
$1,2-3-4$	0.92	0.90	0.83	0.77
$1,3-2-4$	0.89	0.86	0.82	0.78
$2,3-1-4$	0.89	0.88	0.83	0.79
$1,2-4-3$	0.96	0.96	0.83	0.77
$1,4-2-3$	0.94	0.94	0.81	0.77
$2,4-1-3$	0.94	0.94	0.82	0.78
$1,3-4-2$	0.94	0.94	0.83	0.78
$1,4-3-2$	0.93	0.92	0.81	0.73
$3,4-1-2$	0.95	0.95	0.82	0.78
$2,3-4-1$	0.91	0.91	0.80	0.73
$2,4-3-1$	0.91	0.91	0.81	0.73
$3,4-2-1$	0.94	0.94	0.80	0.74
	0.93 $\pm$0.02	0.92 $\pm$0.03	0.82 $\pm$0.01	0.76 $\pm$0.02

explore in future work could be to analyse the area of the image in the extreme of the fitted arms. In the last case, we cannot see any action in this particular frame, while the system indicates a right turn. In fact, in the previous frames, the cyclist actually indicates a right turn; thus, overall this error is more due to the fact that annotating the starting and ending of a given action can have a couple of frames of ambiguity. Therefore, in practice, not detecting any action in this frame or a right turn probably must be considered as correct. Fig. 4.9 shows error cases in the YouTube videos for T=14. From left to right, one case due to having the cyclist arm indicating the action out of the BB, two cases due to a bad fitting of the skeleton because adverse conditions (bag in the back, narrow BBs and low contrast arm-background), and two cases where the action has just started and it is not yet clear enough (*e.g.* the last case is just the starting of the left-turn action correctly classified in the left example of Fig. 4.7).

Finally, we analyse which features are the most important according to the RF

Table 4.3 – Classification accuracy (**Acc**) and **F1** score for $T = 14$, both ranging from 0 to 1. **Train-Val-Test** refer to the cyclist ID of CASR used for training, validation, and testing, respectively. This turns in 12 runs. For each run, we also report generalization results on the annotated YouTube videos (**Acc-YT**, **F1-YT**). The average and standard deviation of each metric is also reported.

Train−Val−Test	**Acc**	**F1**	**Acc-YT**	**F1-YT**
$1,2-3-4$	0.92	0.91	0.85	0.78
$1,3-2-4$	0.89	0.87	0.83	0.79
$2,3-1-4$	0.89	0.87	0.84	0.80
$1,2-4-3$	0.96	0.96	0.85	0.79
$1,4-2-3$	0.94	0.94	0.83	0.79
$2,4-1-3$	0.94	0.94	0.84	0.79
$1,3-4-2$	0.95	0.94	0.86	0.79
$1,4-3-2$	0.93	0.92	0.83	0.74
$3,4-1-2$	0.95	0.94	0.84	0.79
$2,3-4-1$	0.93	0.93	0.81	0.74
$2,4-3-1$	0.93	0.93	0.82	0.75
$3,4-2-1$	0.95	0.95	0.80	0.74
	0.93 $\pm$0.02	0.92 $\pm$0.03	0.83 $\pm$0.02	0.77 $\pm$0.02

classification. Table 4.5 shows the case T=1, choosing the classifier in the 4th row of Table 4.2 (best performing in accuracy terms). We can see that most of the features correspond to angles defined by either a keypoint from neck/shoulders/waist/legs and two keypoints from arms (*e.g.* $\Theta(4,6,7)$, taking Fig. 3.10 as reference), or two keypoints from the former set and one from the later (*e.g.* $\Theta(3,2,7)$). Distances between these two sets are also among the most relevant (*e.g.* $\mathrm{L}(8,7)$). Table 4.6 shows the case T=14, choosing the classifier in the 4th row of Table 4.3 (best performing in accuracy terms). We see that current frame (*i.e.* frame 14 of the temporal sliding window composed of current and past frames) mainly contributes with angle-based futures, which is coherent with the results of T=1; *i.e.* to favour early sign recognition despite using more frames. We see also that there are many distance-based features between neck/shoulder/waist/leg and arm keypoints (*e.g.* $\mathrm{L}^8(8,7), \mathrm{L}^9_x(10,1)$), most of them are concentrated in the middle of the sliding window (from frame 5 to 10 we find 12 feature-based features from the top-25, from frame 12 to 14 we find 6)

Table 4.4 – Average classification accuracy (**Acc**) and **F1** score in CASR, both ranging from 0 to 1. Corresponding results on YouTube videos are also reported as **Acc-YT** and **F1-YT**. Results are reported for noise free keypoints, *i.e.* using them as provided by the skeleton fitting algorithm, as well as for two different levels of noise (20% and 30%) on their location, which is forced at testing time (main text for details).

T=1	**Acc**	**F1**	**Acc-YT**	**F1-YT**
$Noisefree$	0.93 ± 0.02	0.92 ± 0.03	0.82 ± 0.01	0.76 ± 0.02
20%	0.91 ± 0.02	0.90 ± 0.03	0.81 ± 0.01	0.75 ± 0.02
30%	0.87 ± 0.02	0.86 ± 0.03	0.80 ± 0.01	0.72 ± 0.02
T=14	**Acc**	**F1**	**Acc-YT**	**F1-YT**
$Noisefree$	0.93 ± 0.02	0.92 ± 0.03	0.83 ± 0.02	0.77 ± 0.02
20%	0.93 ± 0.02	0.92 ± 0.02	0.83 ± 0.02	0.77 ± 0.02
30%	0.92 ± 0.02	0.92 ± 0.02	0.83 ± 0.02	0.76 ± 0.02

Table 4.5 – For T=1, top-25 most relevant cyclist skeleton-based features from left-to-right and top-to-bottom.

$\Theta(4,6,7)$	$\Theta(9,6,7)$	$\mathrm{L}(8,7)$	$\mathrm{L}(10,7)$	$\mathrm{L}_x(10,1)$
$\Theta(3,6,7)$	$\mathrm{L}_x(10,11)$	$\mathrm{L}(10,6)$	$\Theta(3,2,7)$	$\Theta(6,3,7)$
$\mathrm{L}(3,2)$	$\Theta(11,5,6)$	$\Theta(4,7,6)$	$\Theta(10,6,7)$	$\Theta(5,6,13)$
$\Theta(5,6,12)$	$\Theta(8,6,7)$	$\Theta(11,6,7)$	$\Theta(3,7,6)$	$\Theta(4,2,7)$
$\Theta(6,2,7)$	$\Theta(12,6,7)$	$\Theta(5,6,10)$	$\mathrm{L}(5,9)$	$\Theta(3,6,12)$

which make the sign classification more stable once the cyclist has indicate it during $\sim 200-300$ms. Note also that for cyclists, 25 features correspond to the $\sim 2\%$ for T=1 ove the set available during training, and $\sim 0.15\%$ for T=14.

4.5 Conclusion

In this chapter we have evaluated a monocular vision-based pipeline to address the recognition of cyclist intentions. We have addressed the recognition of cyclist arm signs, in this case elaborating our own dataset, CASR, due to the lack of publicly

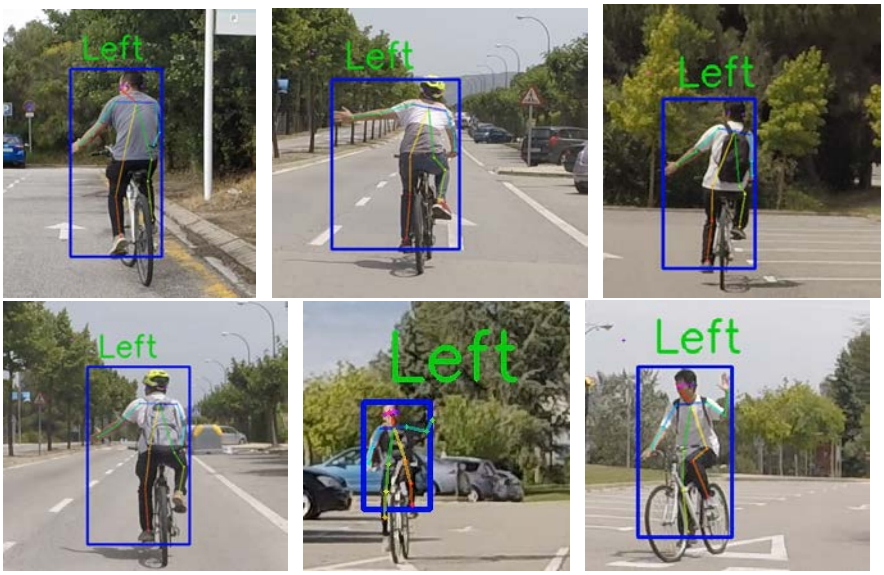


Figure 4.4 – Examples of correct predictions in CASR for cyclist left turn indications (cropped from the original images). Remind that we are using a vehicle-centric criteria, this is why for oncoming cyclist an indication as right-turn must be classified as left-turn.

Table 4.6 – For T=14, top-25 most relevant cyclist skeleton-based features from left-to-right and top-to-bottom.

$L^{14}(8,7)$	$L_x^{12}(10,1)$	$L^{8}(8,7)$	$L_x^{9}(10,1)$	$\Theta^{14}(12,5,6)$
$L_x^{5}(10,1)$	$L_x^{10}(10,1)$	$L_x^{6}(10,1)$	$L^{13}(5,9)$	$L^{6}(8,7)$
$L^{13}(10,6)$	$L_x^{2}(10,1)$	$L_x^{13}(10,1)$	$\Theta^{14}(3,6,2)$	$\Theta^{12}(2,3,7)$
$L_x^{8}(10,1)$	$\Theta^{14}(8,6,7)$	$\Theta^{12}(4,2,7)$	$L_x^{7}(10,1)$	$L^{12}(5,9)$
$L^{8}(5,9)$	$\Theta^{14}(11,5,6)$	$L^{7}(4,11)$	$L_x^{9}(10,11)$	$L^{10}(8,11)$

available ones with annotations for such task. Our work hypothesis is that human skeletons fitted on 2D images already convey a very rich information to perform cyclist intention recognition, *i.e.* neither requiring depth nor optical flow information for the recognition task in itself. The obtained results support our work hypothesis,

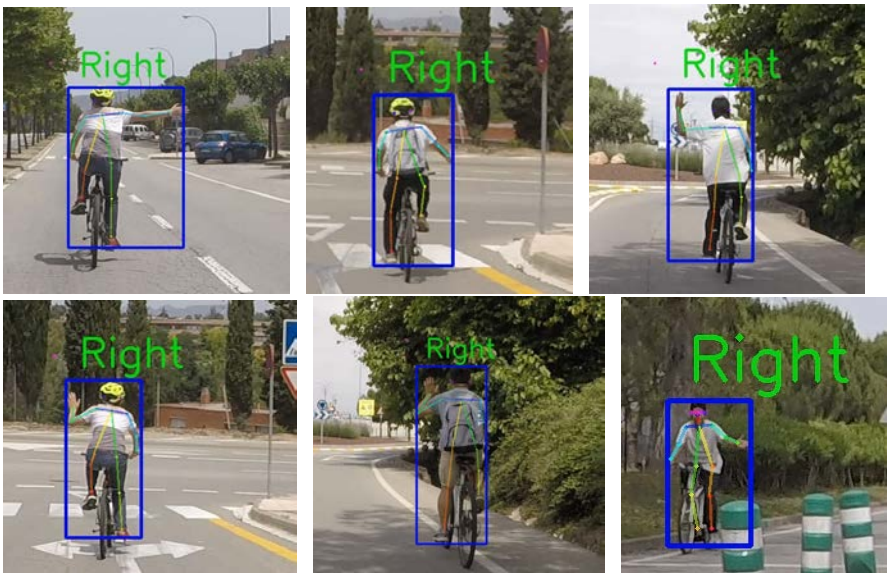


Figure 4.5 – Examples of correct predictions in CASR for cyclist right turn indications (cropped from the original images). Remind that we are using a vehicle-centric criteria, this is why for oncoming cyclist an indication as left-turn must be classified as right-turn.

since by analysing features of the the fitted skeletons over a relatively small temporal sliding window (~ 500ms), the cyclist intention recognition task is performed with a high accuracy. We have showed quantitative results supporting this claim, and we have also brought qualitative results (correct recognition cases, current failure cases, top-features driving recognition) illustrating the reasons. Other researchers can use our approach as part of a modular environment perception pipeline [16], in a similar way than affordances on end-to-end driving models [52], or as additional cue on systems relying on 3D trajectory prediction for recognizing cyclist intentions [32].

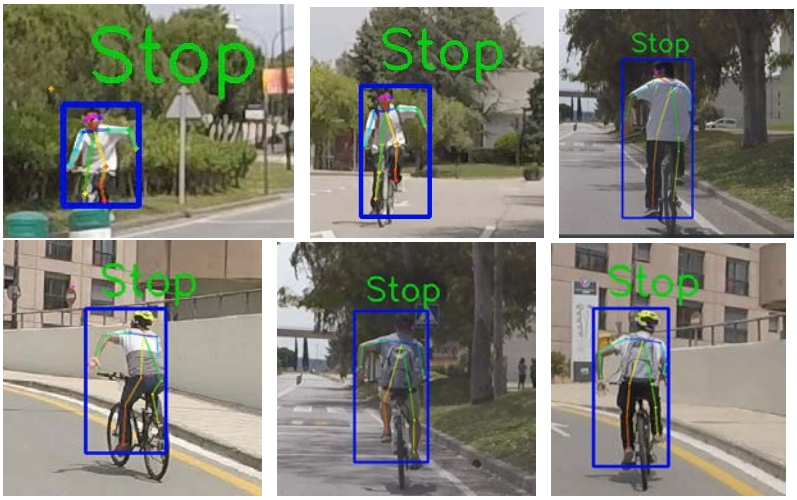


Figure 4.6 – Examples of correct predictions in CASR for cyclist stop indications (cropped from the original images).

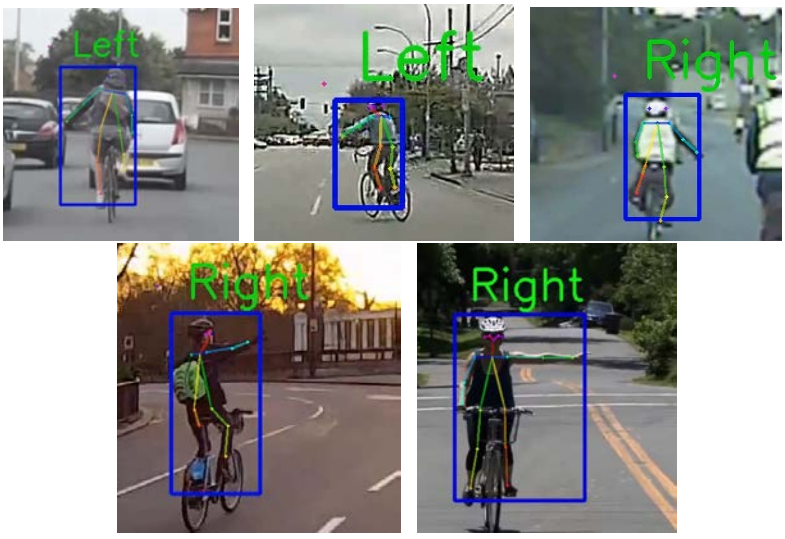


Figure 4.7 – Examples of correct predictions in YouTube images (cropped from the original images).

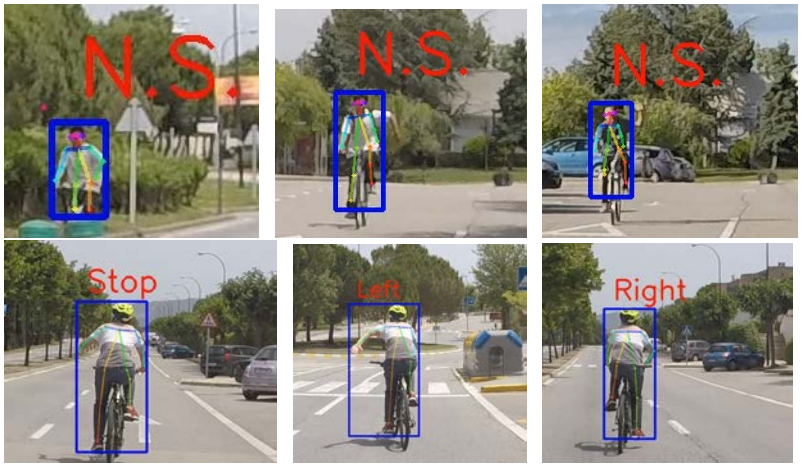


Figure 4.8 – Wrong predictions in CASR for cyclist indications (cropped from the original images). 'N.S.' stands for *no sign*.

Figure 4.9 – Wrong predictions in YouTube images (cropped from the original images). 'N.S.' stands for *no sign*.

5 Conclusions

Anticipating the intentions of vulnerable road users (VRUs) such as pedestrians and cyclists is critical for performing safe and comfortable driving maneuvers in human driving. Therefore, it must be taken into account by advanced driver assistant systems (ADAS) and especially by fully autonomous vehicles (AVs). The latter case, may be even more critical since the current VRU-driver gestural communication will not exists due to the lack of a human driver.

In this PhD work, we hypothesize that temporal dynamics of VRU's skeleton conveys a very rich information to detect the intentions of such VRUs. Until very recently, fitting VRU's skeleton was a problem in itself. Fortunately, as for many other perception tasks, the breakthrough of convolutional neural networks (CNNs) has allowed robust VRU's skeleton fitting even from still monocular images. Therefore, this was the perfect time to assess the validity of our hypothesis. In particular, we have assessed two cases. On the one hand, we have studied the possibility of detecting the intention of pedestrians of crossing or not from a pedestrian area (*e.g.* a sidewalk) to the road in the path of the ego-vehicle. On the other hand, we have studied the possibility of detecting cyclist arm signs as a way to understand their intention. In the former case, pedestrians do not provide explicit indications, while in the latter case cyclist do since is even regulated by traffic rules. In both cases, we want to use an analogous perception pipeline rather than elaborating a very different case for each type of VRU. These goals have been sated in Chapter 1, where we have also put this PhD research in context.

Our initial work in this line is developed in Chapter 2, where we focused on recognizing pedestrian intentions. At the time this work was done, there was only one publicly available dataset for researching in this topic, which we have called Daimler dataset in this PhD. It is not a dataset acquired in naturalistic conditions since the pedestrians involved pedestrians were asked to perform the considered intention-related actions (crossing, stopping, blending and starting); moreover, only a single pedestrian was appearing per frame. Therefore, our focus was to assess if a CNN-based state-of-the-art human skeleton fitting could indeed be the key to recognize pedestrian intentions. This would have the advantage of relying

on a relatively simple perception infrastructure, since such CNN can perform on still monocular images, *i.e.* contrarily to other existing literature which relied on stereo, dense optical flow, and adhoc heuristics to perform pedestrian intention recognition. Indeed, we defined a set of skeleton-based features, aggregated over time, which combined with a shallow classifier (RF and SVM were tested) showed state-of-the-art performance on the Daimler dataset. In particular, our model anticipated by 750ms the intention of keep walking to cross towards the road, by 250ms the intention of bending, and by 187ms the intention of starting to cross when being on a standstill pose.

At the time the work reported in Chapter 2 was finished, it appeared a new publicly available dataset which, after doing some minor adaptations, was ready to support our research on detecting pedestrian intentions. In particular, this dataset, termed as JAAD, was acquired in naturalistic driving and we could easily adapt its ground truth to address the crossing/not-crossing (C/NC) pedestrian action classification task; which we did in Chapter 3. In this case, we integrated a full vision-based perception pipeline consisting of pedestrian detection, tracking and pose estimation (*i.e.* skeleton fitting); allowing to assess the performance of our pedestrian intention recognition classifier in more challenging conditions. Detection and tracking were also based on CNN architectures. The obtained results confirmed our proposal as the state-of-the-art for JAAD dataset. Moreover, we show that the features we defined on top of the fitted skeletons were more effective than others proposed in the literature, which consist on leveraging feature layers from the CNN pedestrian detection architecture. We also analyzed which are the most important features used by our pedestrian intention classifier. We found that they correspond to angles formed by segments that mostly connect keypoints from either shoulder and legs, or shoulder and waist; thus, capturing global pose. Finally, we also showed how our method is relatively robust to noise on the skeleton fitting process, which we did by assessing the its performance when injecting noise to the fitted skeleton keypoints. In fact, this setting revealed the relevance of aggregating skeleton features over multiple frames by a time sliding window procedure.

Once we were satisfied with the results obtained for pedestrian intention recognition, we decided to assess the performance of the same methodology for detecting the intentions of cyclists. In this case, we assumed cyclists respect the traffic rules for indicating their intentions, in other words, they perform arm signs. Therefore, Chapter 4 focuses on cyclist arm sign recognition. Again, one of the main problems was the lack of publicly available datasets prepared for such perception task. In this case, we decided to create our own one. Since our idea is to release it publicly, we decided to start by a similar approach than Daimler for their pedestrian intention dataset. Thus, we agreed with several cyclists to appear in this dataset. We asked them to ride properly indicating their intention of either turning left, or right, or

stopping, whenever required according to the road and the route they follow. We termed this dataset as CASR (cyclist arm sign recognition). We annotated CASR as well as some additional videos found in YouTube for assessing our VRU intention recognition proposal; *i.e.* to validate our hypothesis about the usefulness of human skeleton fitting as key to detect cyclists intentions too. Again, we showed the high accuracy of our method, being able to robustly detecting cyclist arm signs in half a second. As for pedestrian intention recognition, we also assessed which are the most important features. We found that angle-based features (*i.e.* computed on top of the fitted skeleton keypoints) of current frame are the most relevant, complemented by similar ones in next immediate frames as a way of robustifying spurious arm poses resembling cyclist arm signs. An analogous study to the case of pedestrian detection, also confirmed the robustness of our method when injecting noise in the fitted skeleton keypoints. To the best of our knowledge this is the first study on vision-based cyclist arm sign recognition relying on human skeleton pose. We expect that after publication of CASR, other researchers can test their own methods.

Overall, we think that the work presented along Chapter 2, Chapter 3, and Chapter 4, validates our hypothesis: monocular vision-based human pose estimation is a robust information to perform VRU intention recognition. Our proposal can be further evaluated on hopefully new oncoming datasets, or even integrated in methods based on the computation of explicit 3D trajectories of VRUs.

A Appendix

A.1 Detection and classification

A.1.1 Faster R-CNN

As can be seen in Fig. A.1, Faster R-CNN is a convolutional neural network specialized on generic object detection in still images. It consists of a Region Proposal Network (RPN) followed by a Classification Network that determines the content of the generated proposals (which are rectangular image windows). Both networks share the same convolutional layers. The RPN uses a network on top of the convolution layers for extracting features giving rise to different proposals. Regarding the classification network, the features are fed into a box-regression layer (*reg*) and a box-classification layer (*cls*) for training the classifier as shown in Fig. A.2. In Chapter 3, we fine-tuned for pedestrian detection a Faster R-CNN detector [5] which used a VGG [58] pre-trained on ImageNet as convolutional layers. Then we tested the trained detector in JAAD and CASR dataset. Although there is a false positive and two miss detections in Fig. A.3, the model is providing te accuracy we needed for our research. Fig. A.4 shows a right detection result on one of CASR image, but the right arm is out of the bounding box since the pedestrians on JAAD does not have such pose. This actually a problem for performing cycislts arm recognition and the reason to switch to Mask R-CNN for this task.

A.1.2 Mask R-CNN

Mask R-CNN [24] is an extension of Faster R-CNN. It has the same two-stage architecture with a RPN as the first stage. Comparing with Faster R-CNN, in addition to *reg* and *cls*, a binary mask is also predicted in the second stage (see Fig. A.5). Off-the-shelf Mask R-CNN models, based on a ResNet [26] backbone network, already perform relatively well for the task of person detection in still images. For detecting cyclists it was more convenient than Faster R-CNN, since the former is able to frame them even if they indicate traffic signs with open arms, which was not the case of

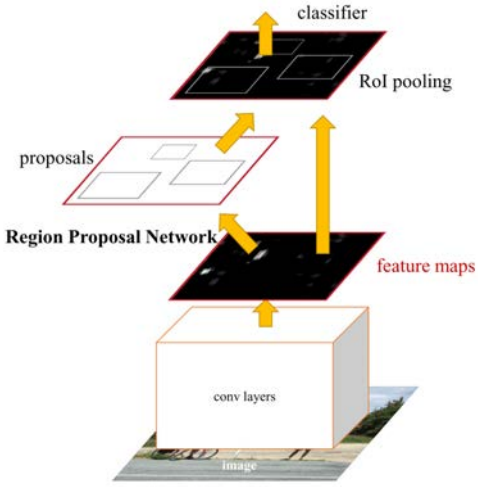


Figure A.1 – Faster R-CNN architecture [50].

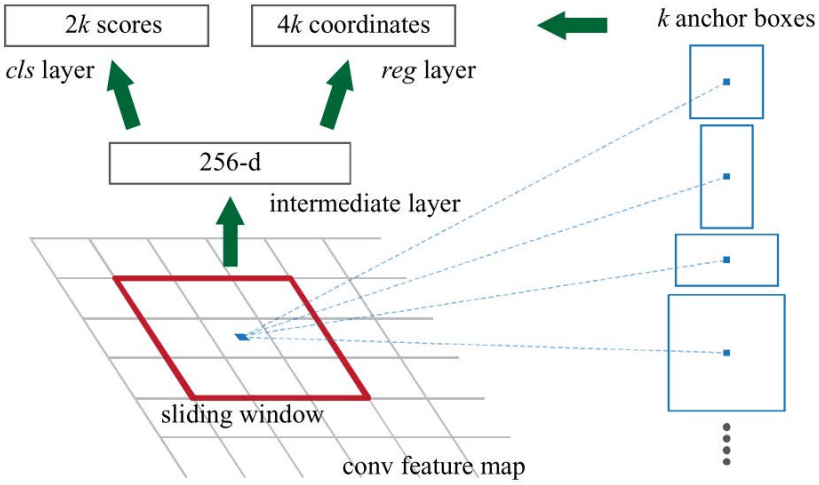


Figure A.2 – Reigion Proposal Network (RPN) [50].

Figure A.3 – Pedestrian detection result by using Faster R-CNN [50] and VGG [58].

Figure A.4 – Cyclist detection result by using Faster R-CNN [50] and VGG [58].

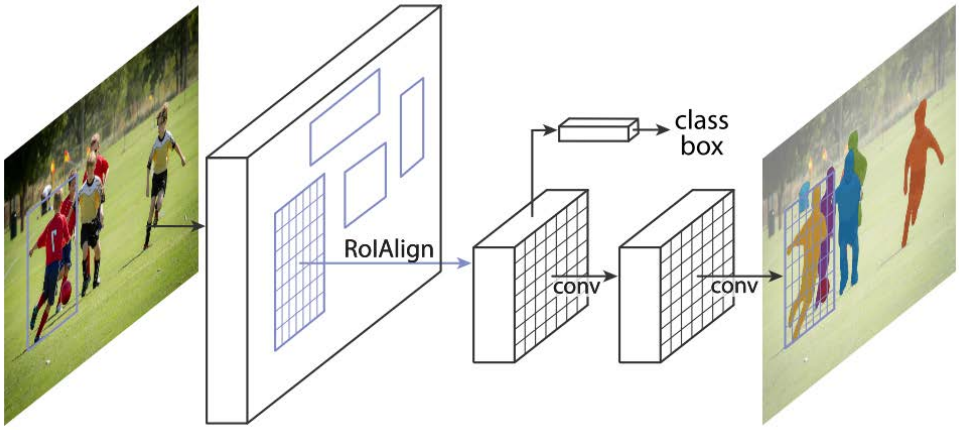


Figure A.5 – Mask R-CNN architecture [24].

Figure A.6 – Cyclist detection result by using Mask R-CNN [24] and ResNet [26]. Compare the bounding box with Fig. A.4.

Figure A.7 – Pedestrian detection result by using Mask R-CNN [24] and ResNet [26].

Figure A.8 – Pedestrian detection result by using Mask R-CNN [24] and Resnet [26].

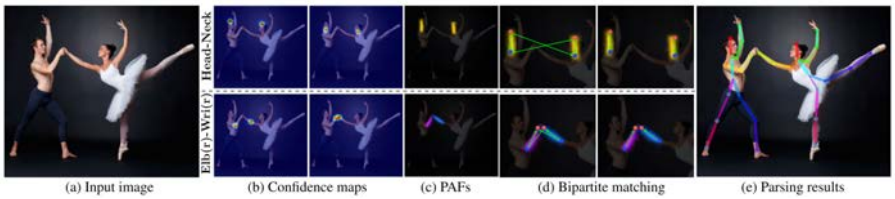


Figure A.9 – Overall pipeline for pose estimation according to [3].

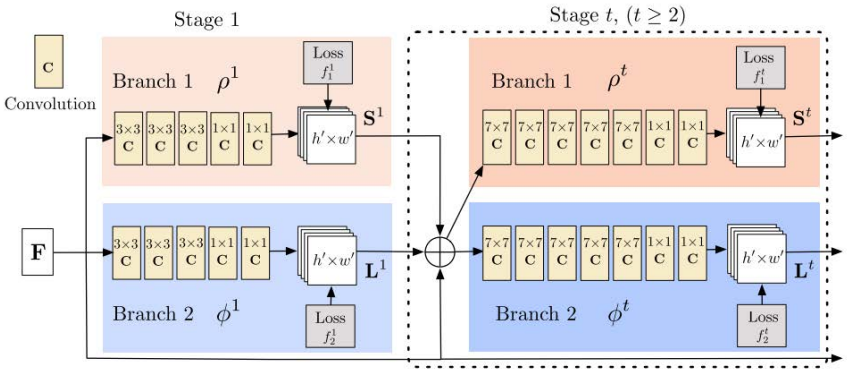


Figure A.10 – Architectural design of the two-branch multi-stage CNN for pose estimation [3]. **F** denotes learned image features from first 10 VGG layers. **L** and **S** denote features after each stage. Each stage in the first branch is predicting confidence score maps, and each stage in the second branch is predicting PAFs. After each stage, the predictions from the two branches together with the image features are concatenated for next stage.

Figure A.11 – Skeleton fitting examples from Cityscapes's images [6] by using OpenPose [3].

the latter. Figures A.6, A.7, and A.8 show detection results from Daimler [55], CASR (see Chapter 4) and Cityscapes [6], respectively.

A.1.3 OpenPose

We use the pose estimation CNN model proposed in [3] since it applies to still 2D images. This CNN model consists of body part detection and part association. We can follow the idea in Fig. A.9. The proposed model takes the entire image as input for a two-branched CNN architecture (see Fig. A.10) to jointly predict confidence maps for body part detection (Fig. A.9b), and part affinity fields (PAFs) for parts association (Fig. A.9c). The parsing step performs a set of bipartite matching to associate body parts candidates (Fig. A.9d) and finally assemble them into full body poses for all persons in the image (Fig. A.9e). Fig. A.11 illustrates two skeleton fitting examples. Even this model can process full images to perform pose estimation without a previous person detection step, in this PhD we first detect persons with more specific networks for the task (such as Faster R-CNN or Mask R-CNN), and then use the windows defined by the detection bounding boxes as input for the pose estimation CNN model. Note that person detection use to be performed anyway on-board, so using the corresponding bounding boxes to focus pose estimation on them is eventually faster and more robust than ignoring such detections.

A.1.4 Random forest

A random forest (RF) ensemble classifier includes several decision trees [2]. Each tree, which consists of split and leaf nodes, is fitted with random sub-samples of the dataset. During the training, split nodes do the binary classification based on the selected features. In testing time, RF takes the average of different trees' prediction as result. In this PhD we have used the scikit-learn package [45] for training and testing with random forest algorithm.

A.1.5 LSTM

Recurrent Neural Networks (RNNs) are models that consider temporal information. Let $x = (x^0, ..., x^{T-1})$ be a sequence of inputs, $h = (h^0, ..., h^{T-1})$ corresponding hidden states and $y = (y^0, ..., y^{T-1})$ the their output states. Then, Equations A.1 and A.2 describe the computation of output and hidden states, respectively. W_{xh}, W_{hh}, W_{ho} represent the connection weights from input (x) to hidden layer (h), hidden (h) to itself, and hidden (h) to output layer (y), respectively. b_h and b_o denote bias vectors, while H($\cdot$) and O($\cdot$) are activation functions.

$$h^t = H(W_{xh}x^t + W_{hh}h^{t-1} + b_h) \tag{A.1}$$

$$y^t = O(W_{h0} h^t + b_o) \tag{A.2}$$

Due to the difficulty of training RNNs (vanishing gradients), LSTMs (long-short term memories) were proposed as a viable alternative. In Fig. A.12, a LSTM cell is shown. It contains the cell c, the input gate i and the forget gate f. The activation functions are described as follows:

$$i^t = \sigma(W_{xi} x^t + W_{hi} h^{t-1} + W_{ci} c^{t-1} + b_i) \tag{A.3}$$

$$f^t = \sigma(W_{xf} x^t + W_{hf} h^{t-1} + W_{cf} c^{t-1} + b_f) \tag{A.4}$$

$$c^t = f^t c^{t-1} + i^t tanh(W_{xc} x^t + W_{hc} h^{t-1} + b_c) \tag{A.5}$$

$$o^t = \sigma(W_{xo} x^t + W_{ho} h^{t-1} + W_{co} c^t + b_o) \tag{A.6}$$

$$h^t = o^t + tanh(c^t) \tag{A.7}$$

A.2 Labeling pedestrians and cyclists

We annotated the TTE (time-to-event) for pedestrians of the JAAD dataset. The event happens in an specific frame, there are two basic cases. On the one hand, the event corresponds to the frame in which the pedestrian decides to start walking towards the road after being stopped in a curbside (Fig. A.13). On the other hand, the event corresponds to the instant when the pedestrian reaches for the first time the border of the sidewalk (curbside) on his/her way from the towards the road surface, he/she can stop (we don't have the annotation) or continue walking (Fig. A.14). Positive values of TTE mean that the event still has not happened, negative values mean that it already passed, and '0' corresponds to the moment when the event is considered to happen. During annotation, we visit forward and backward the sequence around an event for adjusting it properly. We assign the '0' to one

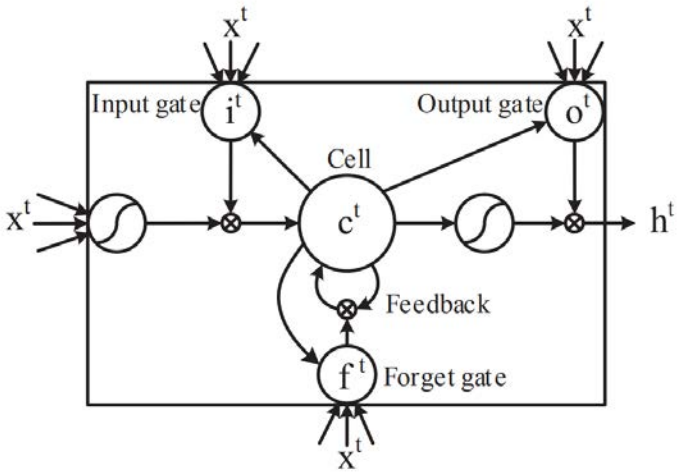


Figure A.12 – Long Short-Term Memory (LSTM) block with one cell [10, 23].

single frame. In our experiments the minimum frame rate is 15 fps, so even if the event happens between frames still the annotation error is below $\sim 33ms$ in average.

Our annotation tool for cyclists was modified based on BeaverDam [57], that is written in Django framework. Fig. A.15 illustrates the framework logic. The annotator interface is shown in Fig. A.16.

Cyclist annotation information consists of an action label, and action state, and cyclist orientation, defined as follows:

- Actions. 0: no signal. -0: turning left. 0-: turning right. |‾‾: stopping.
- Action states. Sta: starting. Hod: holding. End: ending. We do not use this information in the training and testing in this thesis.
- Orientations. It contains 8 different body orientations. We do not use this information in the training and testing in this thesis.

Figures A.17 to A.20 show how we annotated the cyclist videos.

A.3 VRU intention experiments

In Chapter 3 and Chapter 4, we show the quantitative accuracy of intention detection with noise in the keypoints of the skeleton resulting from pose estimation. Fig.

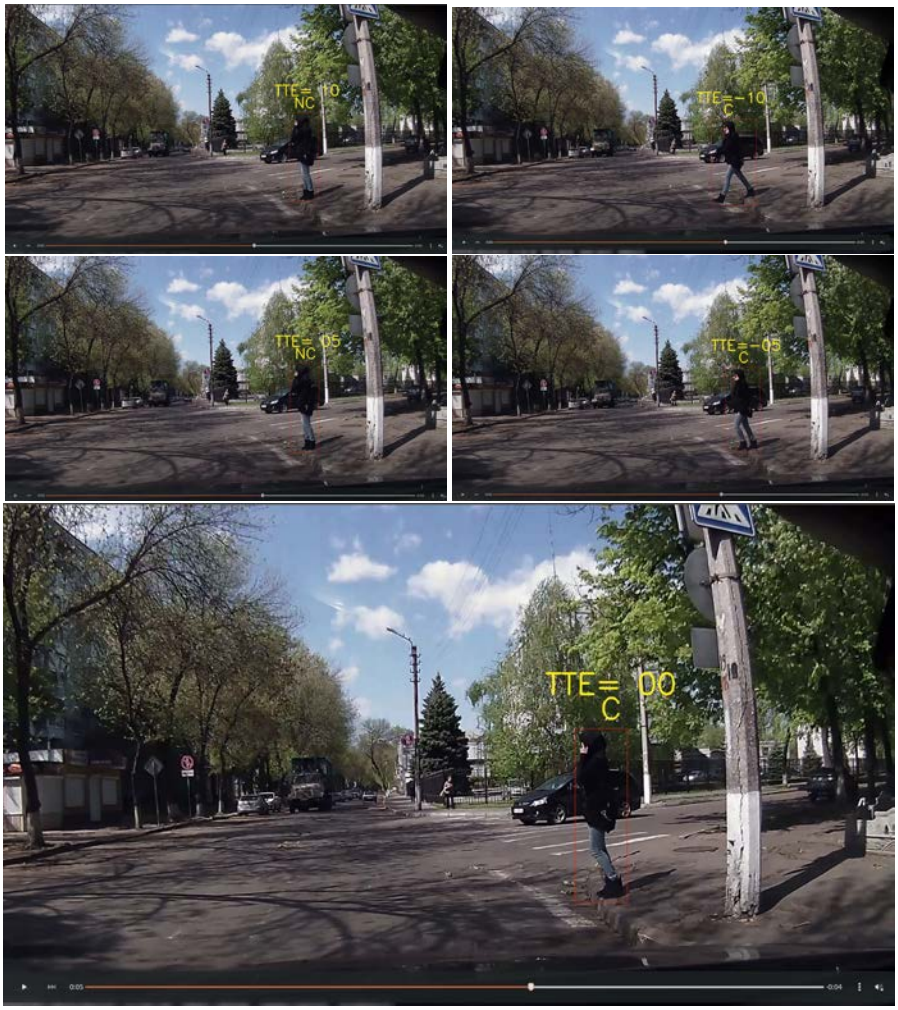


Figure A.13 – Examples of TTE annotation regarding to pedestrian intention recognition in a JAAD sequence. The event corresponds to the value 0, which, in this case, means the pedestrian starts walking after being stopped. Positive values of TTE mean that the event did not yet happen, and negative values mean that it passed.

Figure A.14 – Examples of TTE annotation regarding to pedestrian intention recognition in a JAAD sequence. In this case, the pedestrian keeps walking without stopping at the curbside.

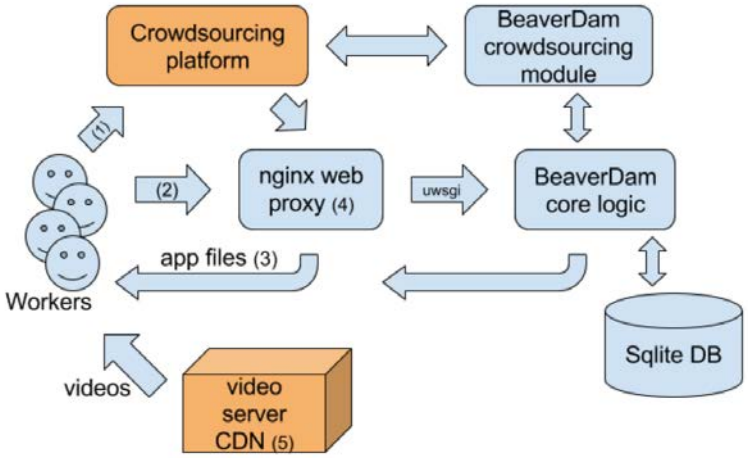


Figure A.15 – BeaverDam's backend server logic. The annotation App is sent in (3). Workers can either be hired through a crowdsourcing platform (1), or hired in-house and use BeaverDam directly (2). The web proxy (4) smoothly handles many requests and forwards static files, and performs HTTPS authentication with HSTS to meet MTurk security requirements. A video server or cloud provider CDN (5) is used to reduce worker download waiting times, a problem of other video labeling tools [57].

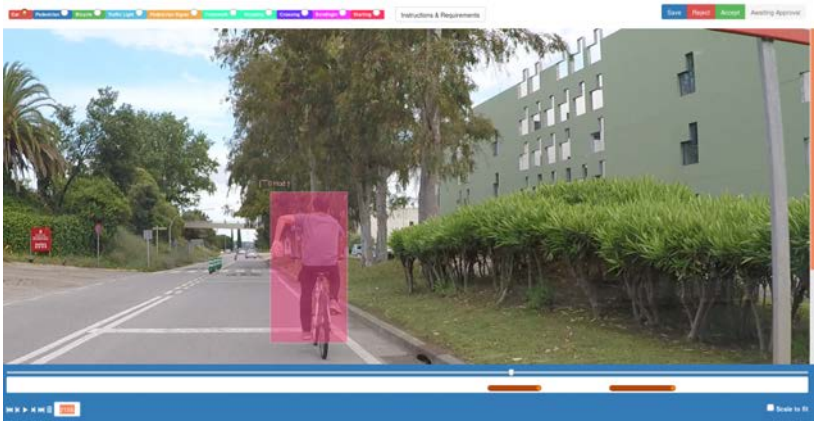


Figure A.16 – Annotator interface.

Figure A.17 – Annotated as no signal and facing forward (body orientation).

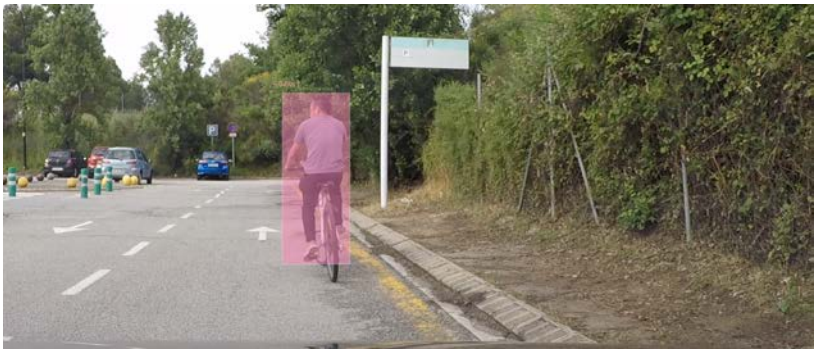


Figure A.18 – Annotated as turning left, starting and facing forward (body orientation).

Figure A.19 – Annotated as turning left, holding and facing forward (body orientation).

Figure A.20 – Annotated as turning left, ending and facing forward (body orientation).

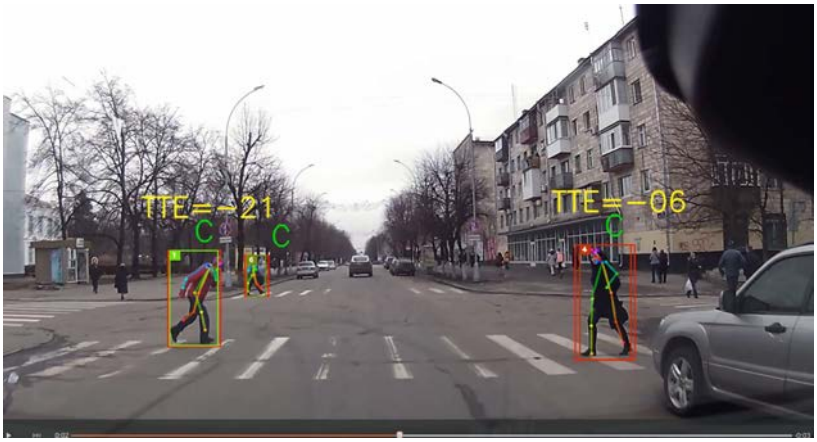


Figure A.21 – A testing result with 30 percent of keypoint noise in JAAD.

Figure A.22 – Turning left detection with 30 percent of keypoint noise.

Figure A.23 – Turning right detection with 30 percent of keypoint noise.

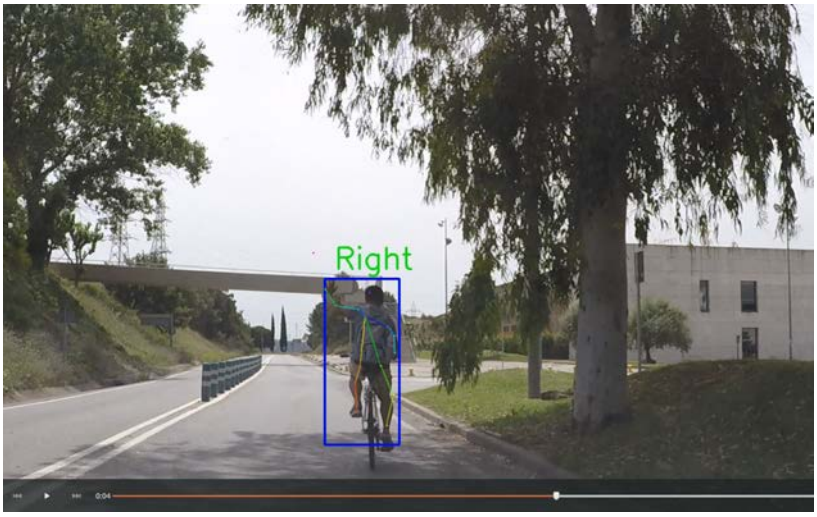


Figure A.24 – Altenative turning right detection with 30 percent of keypoint noise.

Figure A.25 – Stopping detection with 30 percent of keypoint noise.

A.21 illustrates the robustness of our classifier trained in JAAD in this circumstances. From Fig. A.22 to Fig. A.25 show the correct detection of cyclist arm signals also adding noise to the skeleton keypoints.

A.4 Scientific Articles

A.4.1 International Conferences

- **Fang, Zhijie**, and Antonio M. López. "Is the pedestrian going to cross? answering by 2d pose estimation." IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.

A.4.2 Journals

- González, A.; **Fang, Z.**; Socarras, Y.; Serrat, J.; Vázquez, D.; Xu, J.; López, A.M. Pedestrian Detection at Day/Night Time with Visible and FIR Cameras: A Comparison. Sensors 2016, 16, 820.
- **Fang, Z.**; Vázquez, D.; López, A.M. On-Board Detection of Pedestrian Intentions. Sensors 2017, 17, 2193.
- **Fang, Zhijie**, and Antonio M. López. Intention Recognition of Pedestrians and Cyclists by 2D PoseEstimation. IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2019 **(Submitted)**

Bibliography

[1] A.Eldesokey, M. Felsberg, and F.-S. Khan. Ellipse detection for visual cyclists analysis in the wild. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2017.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[5] Xinlei Chen and Abhinav Gupta. An implementation of Faster RCNN with study for region sampling. arXiv preprint arXiv:1702.02138, 2017.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[8] P. Dollár, 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 (T-PAMI)*, 34(4):743–761, 2012.

[9] Piotr Dollár, Serge J Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *British Machine Vision Conference (BMVC)*, 2010.

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015.

[11] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 31(12):2179–2195, 2009.

[12] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 32(9):1627–1645, 2010.

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[14] F. Flohr, M. Dumitru-Guzu, J.F.P. Kooij, and D.M. Gavrila. A probabilistic framework for joint pedestrian head and body orientation estimation. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 16(4):1872–1882, 2015.

[15] Fabian Flohr. *Vulnerable road user detection and orientation estimation for context-aware automated driving*. PhD thesis, Faculty of Science, University of Amsterdam, 2018.

[16] U. Franke. *Computer Vision in Vehicle Technology: land, sea, and air*, chapter Autonomous Driving. Wiley, 2017.

[17] D. Gerónimo and A.M. López. *Vision-based Pedestrian Protection Systems for Intelligent Vehicles*. SpringerBriefs in Computer Science, 2014.

[18] Omair Ghori, Radek Mackowiak, Miguel Bautista, Niklas Beuter, Lucas Drumond, Ferran Diego, and Björn Ommer. Learning to forecast pedestrian intention from pose dynamics. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1277–1284. IEEE, 2018.

[19] Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[21] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[22] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. R-CNNs for pose estimation and action detection. arXiv:1406.5212, 2014.

[23] Alex Graves. *Supervised sequence labelling*, pages 5–13. Springer, 2012.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 346–361. Springer, 2014.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M.J. Black. Towards understanding action recognition. In *International Conference on Computer Vision (ICCV)*, 2013.

[29] C. Keller and D.M. Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 15(2):494–506, 2014.

[30] S. Köhler, M. Goldhammer, S. Bauer, S. Zecha, K. Doll, U. Brunsmann, and K. Dietmayer. Stationary detection of the pedestrian intention at intersections. *IEEE Intelligent Transportation Systems Magazine*, 5(4):87–99, 2013.

[31] S. Köhler, M.Goldhammer, K. Zindler, K. Doll, and K. Dietmeyer. Stereo-vision-based pedestrian's intention detection in a moving vehicle. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[32] Julian FP Kooij, Fabian Flohr, Ewoud AI Pool, and Dariu M Gavrila. Context-based path prediction for targets with switching dynamics. *International Journal of Computer Vision (IJCV)*, pages 1–24, 2018.

[33] Henrik Kretzschmar and Jiajun Zhu. Cyclist hand signal detection by an autonomous vehicle, April 2015. US Patent 9,014,905.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] J.-Y. Kwak, E.-J. Lee, B. Ko, and M. Jeong. Pedestrian's intention prediction based on fuzzy finite automata and spatial-temporal features. In *International Symposium on Electronic Imaging – Video Surveillance and Transportation Imaging Applications*, 2016.

[36] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168. IEEE, 2011.

[37] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D.M. Gavrila. A new benchmark for vision-based cyclist detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[38] X. Li, L. Li, F.B. Flohr, J. Wang, X. Xiong, M. Bernhard, S. Pan, D.M. Gavrila, and K. Li. A unified framework for concurrent pedestrian and cyclist detection. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 18(2):269–281, 2017.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C.L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[40] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[41] M.-M. Meinecke, M. Obojski, D.M. Gavrila, E. Marc, R. Morris, M. Töns, and L. Lettelier. Strategies in terms of vulnerable road users. EU Project SAVE-U, Deliverable D6, 2003.

[42] L. Neumann and A. Vedaldi. Tiny people pose. In *Asian Conference on Computer Vision*, 2018.

[43] World Health Organization et al. Global status report on road safety 2018. Technical report, World Health Organization, 2018.

[44] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision (IJCV)*, 38(1):15–33, 2000.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

[46] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[47] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *International Conference on Computer Vision (ICCV), Workshop*, 2017.

[48] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[49] E. Rehder, H. Kloeden, and C. Stiller. Head detection and orientation estimation for pedestrian safety. In *IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[50] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[52] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments. arXiv:1806.06498, 2018.

[53] Sarah Schmidt and B Färber. Pedestrians at the kerb–recognising the action intentions of humans. *Transportation research part F: traffic psychology and behaviour*, 12(4):300–310, 2009.

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

[55] N. Schneider and D.M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In *German Conference on Pattern Recognition (GCPR)*, 2013.

[56] A. Schulz and R. Stiefelhagen. Pedestrian intention recognition using latent-dynamic conditional random fields. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[57] Anting Shen. Beaverdam: Video annotation tool for computer vision training labels. Master's thesis, University of California, Berkeley, 2016.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3548, 2017.

[60] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.

[61] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto. A data-driven approach for pedestrian intention estimation. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

[62] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision (IJCV)*, 103(1):60–79, 2013.

[63] Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 710–724. Springer, 2008.

[64] Thomas Winkle. Safety benefits of automated vehicles: Extended findings from accident research for development, validation and testing. In *Autonomous Driving*, pages 335–364. Springer, 2016.

[65] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing (ICIP)*, 2017.

[66] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. Towards reaching human performance in pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, May 2017.

[67] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision (ECCV)*, 2016.

[68] Silvia Zuffi, Oren Freifeld, and Michael J Black. From pictorial structures to deformable structures. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3553. IEEE, 2012.