Thèse n° 10 072

Reshaping Perception for Autonomous Driving with Semantic Keypoints

Présentée le 6 mai 2022

Faculté de l'environnement naturel, architectural et construit
Intelligence Visuelle pour les Transports
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Lorenzo BERTONI

Acceptée sur proposition du jury

Prof. N. Geroliminis, président du jury
Prof. A. M. Alahi, directeur de thèse
Prof. M. Cord, rapporteur
Dr J. Shotton, rapporteur
Prof. J.-Ph. Thiran, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

For Maria

Acknowledgements

Firstly, I must thank my supervisor, Alexandre Alahi, for his constant guidance and precious feedback. I am especially grateful for his support in pursuing independent research ideas and encouragement to always push the boundaries.

I would also like to thank Sven Kreiss, a friend and a mentor, who taught me how to be a researcher and a better programmer. Sven, I could not have asked for finer support and advice throughout the years. Working with you has been a terrific pleasure.

I would like to express my special recognition to the members of my thesis committee, Dr. Jamie Shotton, Prof. Matthieu Cord, Prof. Jean-Philippe Thiran, and Prof. Nikolas Geroliminis for their effort in reviewing this dissertation and for their meaningful comments.

I would also like to thank all the excellent researchers I have been fortunate to collaborate with, in particular Taylor Mordan, for his advice and marvelous attention to detail, and Younes Belkada for his contagious enthusiasm. This work would not have been possible without them and the great work of Wenlong Deng, Weijiang Xiong, and Romain Caristan.

Very special gratitude goes to George Adaimi, a fellow student but also the cornerstone of the laboratory. Thanks for your great deal of help with all the GPU-related issues.

I do wish to extend my thank you to my excellent lab friends who helped and supported me through my studies. In particular, thank you: Parth Kothari, for being the best office mate one could ever ask for; Yuejiang Liu for the inspiring discussions and brainstorming sessions; Brian Sifringer, Saeed Saadatnejad, and Hossein Bahari for the endless time spent reviewing drafts.

Last, but by no means least, my heartfelt thanks to my family for their never-faltering encouragement and love.

Lausanne, 24 February 2022 L. B.

Abstract

The field of artificial intelligence is set to fuel the future of mobility by driving forward the transition from advanced driver-assist systems to fully autonomous vehicles (AV). Yet the current technology, backed by cutting-edge deep learning techniques, still leads to fatal accidents and does not convey trust. Current frameworks for 3D perception tasks, such as 3D object detection, are not adequate as they (i) do not generalize well to new scenarios, (ii) do not take into account measures of confidence in their predictions, and (iii) are not suitable for large-scale deployment as mainly based on costly LiDAR sensors.

This doctoral thesis aims to study vision-based deep learning frameworks that can accurately perceive the world in 3D and generalize to new scenarios. We propose to escape the pixel domain using semantic keypoints, a sparse representation for every object in the scene containing meaningful information for 2D and 3D reasoning. The low-dimensionality enables downstream neural networks to focus on essential elements in the scene and improve their generalization capabilities. Furthermore, driven by the limitation of deep learning architectures outputting point estimates, we study how to estimate a confidence interval for each prediction. In particular, we emphasize vulnerable road users, such as pedestrians and cyclists, and explicitly address the long tail of 3D pedestrian detection to contribute to the safety of our roads. We further show the efficacy of our framework on multiple real-world domains by (a) integrating it in an existing AV pipeline, (b) detecting human-robot eye contact in real-world scenarios, and (c) helping verify the compliance of safety measures in the case of the COVID-19 outbreak. Finally, we publicly release the source code of all our projects and develop a unified library to contribute to an open science mission.

Keywords: Autonomous Vehicles, Computer Vision, 3D Perception, 3D Object Detection, Human Pose Estimation, Eye Contact Detection, Action Recognition, Social Distancing

Riassunto

Il campo dell'intelligenza artificiale consentirà di trasformare la mobilità del futuro, permettendo la transizione da sistemi di guida assistita a veicoli completamente autonomi. Attualmente però, i veicoli autonomi, che si basano sulle più moderne tecniche di deep learning, sono ancora causa di incidenti e non trasmettono sicurezza. Un motivo è che le soluzioni di computer vision, come il rilevamento di persone e veicoli in 3D, non sono ottimali, in quanto (i) non generalizzano in nuove situazioni, (ii) non tengono conto di misure di incertezza nelle loro previsioni, e (iii) non sono adatte ad essere implementate su larga scala poichè richiedono costosi sensori LiDARs. Questa tesi di dottorato si pone l'obiettivo di studiare algoritmi di deep learning che, partendo da un'immagine, possano percepire il mondo in 3D accuratamente e che siano in grado di generalizzare. La nostra soluzione consiste nel trasformare un'immagine in Semantic Keypoints: un set limitato di punti che indichino le caratteristiche e la posizione di ogni oggetto di interesse nell'immagine. La bassa dimensionalità permette alle reti neurali a valle di concentrarsi sugli elementi essenziali dell'immagine e migliora la loro capacità di generalizzazione. Inoltre, sviluppiamo reti neurali che al posto di stime puntuali forniscano intervalli di confidenza, e siano quindi più affidabili. Gli algoritmi proposti si focalizzano su soggetti vulnerabili, come pedoni e ciclisti, e ne stimano la distanza, le dimensioni e la direzione nello spazio, per contribuire a rendere i veicoli autonomi più sicuri. In aggiunta analizziamo l'efficacia dei nostri algoritmi in diverse applicazioni pratiche, (a) integrandoli nel software principale di alcuni veicoli autonomi, (b) usandoli per rilevare il contatto visivo tra veicoli e pedoni, e (c) aiutando a verificare la conformità delle misure di sicurezza dovute all'epidemia di COVID-19. Infine, abbiamo rilasciato pubblicamente il codice sorgente di tutti i nostri progetti per contribuire ad una ricerca scientifica aperta a tutti.

Parole chiave: Veicoli Autonomi, Computer Vision, Deep Learning, 3D Perception, 3D Object Detection, Contatto Visivo, Riconoscimento delle Azioni, Distanziamento Sociale

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

We are living in an epoch of transition as artificial intelligence (AI) applications are quickly changing the world as we know it. One of AI's greatest promises is to shape the future of mobility with autonomous vehicles (AVs). The global adoption of AVs would create a social change comparable to the invention of the automobile itself [71]. AVs have the potential to increase mobility for the elderly or those with disabilities, to deliver goods in a cost-effective way, and above all, to save lives. Car accidents are the leading cause of death in people under 30 [137], and 1.25 million people die every year in a car crash. Crucially, 94% of road accidents are caused by human error [190] and they are all considered to be preventable. AVs would offer all the advantages of cars without their drawbacks, and the industry has already invested $120 billion in mobility start-ups from 2017 to 2019 alone [75]. However, despite remarkable progresses from academia and industry alike, integrating AVs into our society remains a grand challenge yet to be solved.

At their core, AVs lack a decision-making software able to generalize to new situations [71]. Research has been shifting from the robotic sense-plan-act paradigm with hand-crafted rules [179] to deep learning approaches that go beyond the driving rule books. However, AVs still do not understand the world in the way that a human being does. Their software is trained to recognize objects or imitate human behavior by using large datasets of real-world driving conditions. Nevertheless, data accumulated over millions of kilometers of driving does not imply that all the critical situations are covered. A human who learned to drive in Rome or Paris is reasonably expected to carry on in other European cities, while AVs do not generalize to new environments. They are often being tested in well-designated areas or specific neighborhoods [93], and multi-city tests are still constituting a challenge [117].

One of the reasons why generalized self-driving remains a grand challenge is that AVs need to accurately perceive the world in 3D. The field of computer vision has made terrific advances in 2D tasks such as image classification [97, 96], object detection [177, 160] and 2D human pose

(a) Samples from the ImageNet dataset [47], where the goal is to classify what category each image belongs to.

(b) A crowded scene recorded from an autonomous vehicle in the city of London, where the goal is to detect the distance from the camera, the orientation, and 3D dimensions of each pedestrian in the scene. The numbers represent the radial distance of each pedestrian from the ego camera, an attribute that cannot be deducted by just looking at the image. Estimates are obtained using our MonoLoco algorithm [23] described in Chapter 3. The image resolution is limited by run time constraints.

Figure 1.1 – Showing the evolution of computer vision tasks with deep learning; in this case, from 2D image classification (a) to 3D object detection (b).

estimation [178, 34], but the performances of the corresponding 3D tasks, such as 3D object detection [39, 40] or 3D pose estimation [122], are still lagging behind. Two main factors impact the performances of deep-learning-based systems for 3D tasks. On the one hand, perceiving the world in 3D from images is more complex than working directly with its 2D projection. This

is especially true for autonomous driving scenarios with no control over the environment or the distance of pedestrians. In Figure 1.1, we visualize samples from a popular dataset for image classification, ImageNet [47], and we compare them with a real-world image captured from an AV in the city of London. In the former case, the task is to classify the image according to its class. In the latter case, the task is 3D object detection, where each dynamic object is detected in 3D, *i.e.*, the 3D world coordinates, orientation, and dimensions. In particular, estimating the 3D location of objects requires an understanding of the geometry of the object and the surrounding scene (*e.g.*, is a car smaller or just further away?). Recovering 3D structures from the projected images of moving objects or scenes is an ambiguous task, which introduces a new set of challenges not encountered in 2D perception tasks.
On the other hand, deep-learning systems that perceive the world in 3D are usually trained with smaller datasets. The ImageNet dataset contains over 14 million labeled images, while the pioneering dataset for 3D object detection, KITTI [64] only includes around 7500 images, even if released four years after ImageNet. Annotating a 3D dataset is not only more time-consuming, as every object in the 3D scene needs to be labeled, but it also requires collecting ground-truth data with specialized hardware such as LiDARs. A human annotator can easily distinguish the types of animals in Figure 1.1a, but cannot estimate how far people are from the camera in Figure 1.1b just by looking at the image.
The complexity of the 3D world and the high cost of 3D labels make it harder for 3D perception systems to perform on pair with 2D perception ones and to generalize well to unseen scenarios.

1.2 Approach

This thesis aims to improve vision-based 3D perception for autonomous driving by designing a pipeline suitable for small training data and effective for cross-dataset generalization. We suggest dealing with the 3D world complexity by leveraging the exceptional progress of computer vision for 2D tasks. More specifically, we propose not to reason in the pixel domain but escape it using keypoints, a sparse representation for every object in the scene. We define the following term:

Definition 1.2.1 (Semantic Keypoints). *Semantic keypoints are spatial points associated with high-level attributes, which convey meaningful information for 2D and 3D reasoning when grouped together.*

Examples of semantic keypoints are “left shoulders” of humans, or “left brake lights” of vehicles, or “tails” of animals. They indicate the spatial location in the image of a specific part while conveying information about it, *e.g.*, indicating that the point corresponds to a right knee or that the brake light is on. Semantic keypoints are also associated with a specific instance, *e.g.*, a person or a vehicle, enabling connections between them. They are opposed to keypoints used in classical feature detectors that focus on the local geometry of the pixel intensities, like “corners” and “edges” [9, 18]. Figure 1.2 compares keypoints obtained from the open-source ORB algorithm [166] with the semantic keypoints obtained from our OpenPifPaf algorithm [94] presented in Chapter 2.

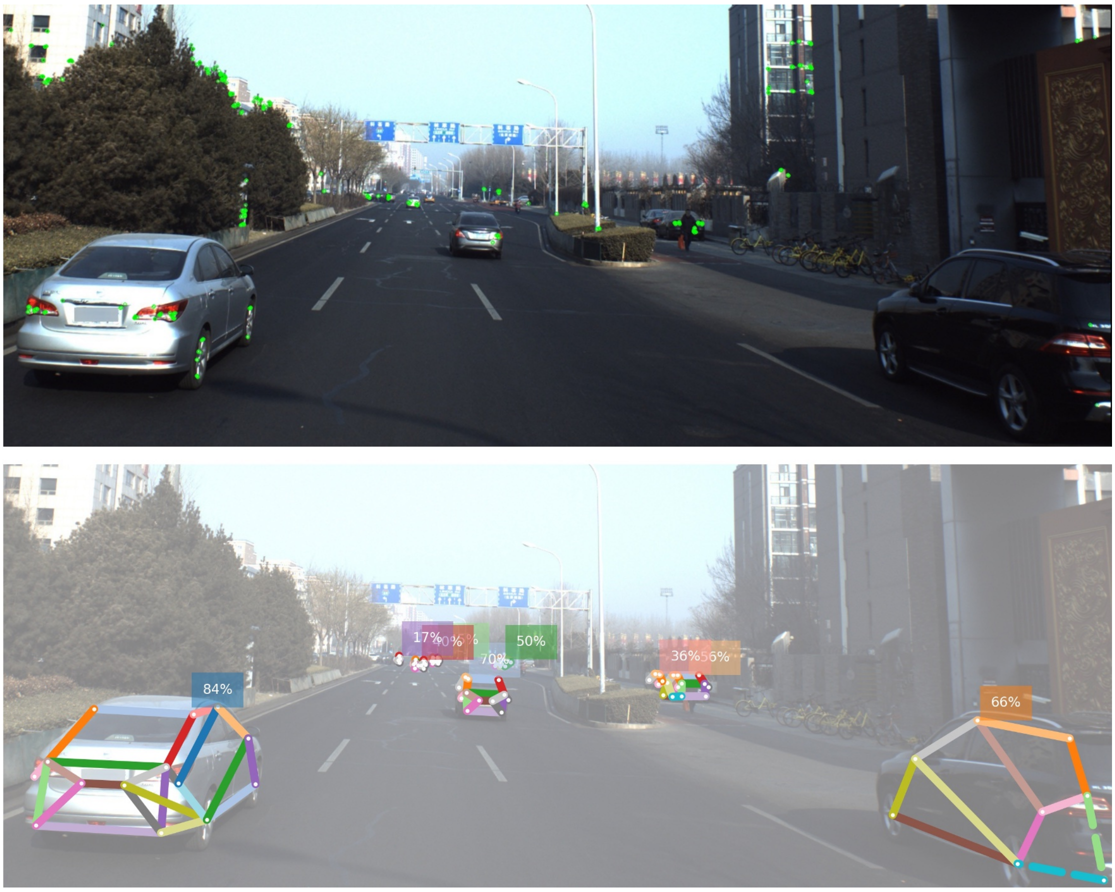


Figure 1.2 – Visually comparing keypoints used in classical feature detectors with semantic keypoints. In the top image, we show the keypoints obtained with the ORB algorithm [166]. They represent interest points that are expressive in texture; *e.g.*, an intersection point between two or more edge segments. In the bottom image, we visualize the semantic keypoints obtained with our OpenPifPaf algorithm (described in Chapter 2). They represent a specific part of an object, *e.g.*, the center of the front-left wheel, and they are linked between them with unique connections represented with different colors, *e.g.*, front-left wheel and left side-view mirror. Each keypoint is also associated with a specific instance; in this case, a vehicle.

Our intuition is that this low-dimensional representation can simplify perceptions tasks by adding inductive biases on the essential elements to focus on. Simultaneously, it can improve the generalization capabilities of perception systems by filtering our redundant information. For instance, when people are crossing the street, they may look and dress differently across the world while sharing the same set of semantic keypoints. Such representation can therefore help AVs to better generalize to new cities. Figure 1.3 shows the semantic keypoints representation of people walking in EPFL campus. Information regarding people orientation and dimensions is preserved while providing invariance to many factors, including background scenes, lighting, textures and clothes. Additionally, semantic keypoints are a sparse representation of the scene, making them ideal for collecting large-scale datasets of labeled examples. In the case of human keypoints, the popular COCO dataset [111] contains 1.7 million labeled keypoints over 200,000 images.

Our objective is to detect a set of 2D semantic keypoints for every instance in an image and use these keypoints to perform 3D reasoning through a neural network. First, we show that the concept of semantic keypoints can be applied to any object by developing a human pose detector and extending it to vehicle and animal categories. Second, we validate the effectiveness of keypoints for 2D and 3D perception tasks by focusing on pedestrians, a challenging yet critical category in the context of autonomous driving.

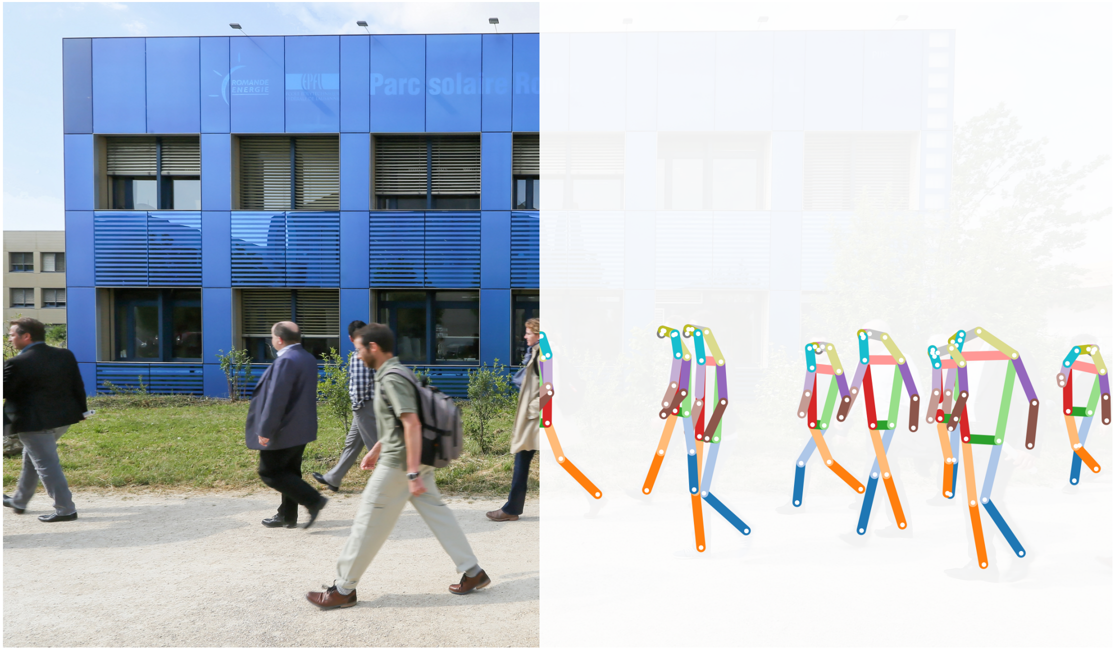


Figure 1.3 – Representing humans with semantic keypoints - a low dimensional representation that contains enough information for 2D and 3D reasoning. A tour of EPFL campus, ©Alain Herzog / EPFL

This thesis proposes vision-based methods that perceive humans in the 3D world. One of the critical challenges in the autonomous driving context is to locate humans surrounding a vehicle. Hence, we focus on the 3D pedestrian detection task: estimating the 3D location, orientation, and 3D dimensions of every pedestrian in the scene. We specifically tackle the long tail of detections by making our framework more robust to outliers and estimating confidence intervals rather than point estimates. Further, we expand our scope to different tasks. We study the impact of semantic keypoints in understanding atomic actions (walking, bending, sitting, etc.) and communicative activities (*e.g.*, do pedestrians pay attention to the incoming traffic?). In addition, we investigate the role of semantic keypoints in real-world applications for autonomous driving and beyond. We specifically focus on testing how our algorithms generalize to unseen scenarios. Finally, inspired by the Workshop on Insights from Negative Results in NLP [173], we conclude some of our chapters with studies and experiments that did not have positive results but that we believe are helpful to be shared with future researchers.

Our mission is to develop a system that can be deployed ubiquitously in a fleet of autonomous vehicles worldwide and contributes to improving the safety of vulnerable road users. Thus, we aim for a design that excels in generalizing to new situations and can work with low-cost sensors. We direct the attention to vision-based methods that perceive the world with cost-effective monocular or stereo RGB cameras. In contrast to LiDARs, cameras are already installed in a large number of vehicles, and they are orders of magnitude cheaper. If extracting 3D information from a single RGB image is a fundamentally ill-posed problem, there is no intrinsic limitation when using stereo cameras. However, vision-based 3D object detection algorithms are still lagging behind LiDAR ones [197, 210]. These results depend not only on the sensor quality, but they are also a consequence of data availability. In fact, many recent 3D vision datasets include in their sensor suite multiple expensive LiDARs while lacking a low-cost stereo camera [31, 198, 88]. We hope our research can contribute to revert this tendency towards a more balanced sensor suite.

1.3 Thesis Contributions

This thesis aims to show that semantic keypoints are a powerful representation to perceive humans in the 3D world. We establish evidence of this statement by (i) obtaining state-of-the-art results in various 2D and 3D perception tasks, (ii) testing our methods on real-world autonomous driving applications, and (iii) extending our design beyond autonomous driving applications by measuring social distancing to help fight COVID-19. The key contributions are:

Semantic Keypoints. We suggest semantic keypoints as an intermediate representation for perception tasks and propose a novel bottom-up pose detector for human pose estimation tailored for AVs. We show that our detector can generalize to car and animal poses, demonstrating its suitability as a holistic perception framework.

Monocular 3D Pedestrian Detection. We propose a novel method that estimates 3D coordinates given 2D semantic keypoints, outperforms state-of-the-art results, and exhibits cross-dataset generalization properties. We address the ambiguity in the task by predicting meaningful confidence intervals in place of point estimates.

Stereo-based 3D Pedestrian Detection. We propose a novel unified learning framework that leverages the strengths of both monocular and stereo cues to address challenging instances, such as occluded and far-away pedestrians. Our method goes beyond providing state-of-the-art results on standard metrics and shows reasonable estimates even for outliers while communicating confidence measures.

Human-robot Eye Contact Detection. We achieve state-of-the-art results in detecting human-robot eye contact by only using semantic keypoints as input. We tackle real-world scenarios for autonomous vehicles, with no control over the environment or the distance of pedestrians. We also release a large-scale dataset for human-robot eye contact detection that focuses on diverse and unconstrained scenarios for real-world generalization.

Social Interactions and Social Distancing. We extend our neural network architecture to infer humans' social interactions and monitor social distancing to fight the COVID-19 outbreak. We develop a probabilistic approach to detect F-formations and show that our design (i) is privacy-safe, (ii) works with any fixed or moving cameras, and (iii) does not rely on ground plane estimation.

Open-source Software. To improve the openness and reproducibility of research, we released the source code[I] of each of our projects and created a unified 3D vision library from 2D semantic keypoints[II].

1.4 Thesis Structure

The body of this thesis is organized into eight chapters, the first of which is this introduction.

Chapter 2 focuses on how to extract semantic keypoints from raw images. It proposes a novel bottom-up pose detector for human pose estimation based on composite fields. The detector specifically addresses challenges that arise in the context of AVs: (i) crowded and occluded scenes with limited resolution on humans, (ii) robustness to real-world variations like lighting, weather, and occlusions, and (iii) speed for real-time predictions. The chapter also highlights how semantic keypoints are not limited to humans by estimating keypoints of cars and animals. The work in this chapter has been led by Dr Sven Kreiss, and it condenses two publications: a CVPR'19 [95] and a T-ITS'21 [94] one.

Chapter 3 leverages semantic keypoints to tackle the ill-posed problem of 3D pedestrian detection from monocular RGB images. It proposes a deep learning approach that estimates each pedestrian's 3D location, orientation, and bounding box dimensions in the scene. Driven by the limitation of neural networks outputting point estimates, it addresses the ambiguity in the task by predicting confidence intervals through a loss function based on the Laplace distribution. This chapter expands on the paper we presented at ICCV'19 [23].

Chapter 4 shows how monocular and stereo visions solutions for 3D localization are usually developed independently and have their respective strengths and limitations. It proposes a novel unified learning framework that leverages the strengths of both monocular and stereo cues for

[I] Code: https://github.com/vita-epfl/monoloco
[II] Library Video: https://www.youtube.com/watch?v=O5zhzi8mwJ4

3D pedestrian localization. It specifically focuses on outliers and challenging instances, such as occluded and far-away pedestrians. It also compares stereo-based disparity estimation solutions with structure from motion ones. Finally, it critically reviews the official KITTI 3D metrics [64] and proposes a practical 3D localization metric tailored for humans. This chapter expands on the paper accepted at ICRA'21 [25] and includes parts of a previous paper presented at ICRA'20 [48].

Chapter 5 focuses on real-world applications of our 3D pedestrian detector for autonomous driving. On one side, the chapter combines the developed design with a fully learned end-to-end motion planner to improve AVs' intervention rates due to pedestrian interactions. It compares the framework with a data-centric approach and shows how a better data curriculum is more effective in improving end-to-end network performances. On the other side, it tests our keypoint-based architecture on new scenarios and uses it to create 3D pseudo-labels of crowded scenes in London, highlighting the practical value of our research. This chapter results from an applied research project in collaboration with the autonomous driving company Wayve.

Chapter 6 goes beyond 3D pedestrian detection and investigates the effectiveness of semantic keypoints for a downstream 2D task: determining whether pedestrians are paying attention to the incoming vehicles in the street. It defines a light-weight feed-forward neural network that estimates human-robot eye contact detection in the wild given semantic keypoints as inputs. The method achieves state-of-the-art results and conveys better generalization properties than leveraging raw images in an end-to-end network. It also describes the release of a large-scale dataset for human-robot eye contact detection in the wild to study domain adaptation. We have published preliminary results to ArXiv [22] and our work is under review.

Chapter 7 investigates the benefits of semantic keypoints for real-world applications beyond autonomous driving. It extends our framework and develops probabilistic rules to analyze social interactions among humans. It then adapts the method to verify the compliance of recent safety measures due to the COVID-19 outbreak. It also shows that it is possible to redefine the concept of social distancing to go beyond a single measure of distance while preserving the privacy of its users. This Chapter is based on our T-ITS'21 publication [24].

Finally, this thesis ends with concluding remarks. Experimental results have supported our initial claim:

"*Semantic keypoints are an effective representation to perceive humans in the 3D world.*"

1.5 Related Publications

This thesis is based on the material published in the following papers:

- Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *MonoLoco: Monocular 3D Pedestrian Localization and Uncertainty Estimation*. International Conference on Computer Vision

(ICCV), 2019

- Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi, *Pifpaf: composite Fields for Human Pose Estimation*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019
- Wenlong Deng, Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *Joint Human Pose Estimation and Stereo 3D Localization*, The IEEE International Conference on Robotics and Automation (ICRA), 2020
- Lorenzo Bertoni, Sven Kreiss, Tayor Mordan, and Alexandre Alahi, *MonStereo: When Monocular and Stereo Meet at the Tail of 3D Human Localization*, The IEEE International Conference on Robots and Automations (ICRA), 2021
- Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi, *OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association*, IEEE Transactions on Intelligent Transportation Systems, 2021
- Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *Perceiving Humans: from Monocular 3D Localization to Social Distancing*, IEEE Transactions on Intelligent Transportation Systems, 2021
- Younes Belkada⋆, Lorenzo Bertoni⋆, Romain Caristan, Taylor Mordan, and Alexandre Alahi. *Do Pedestrians Pay Attention? Eye Contact Detection in the Wild*, under review, 2021

⋆ denotes equal contributions

2 Semantic Keypoints Detection

This chapter is based on the articles:

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi, *Pifpaf: Composite fields for human pose estimation*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019

Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi, *OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association*, IEEE Transactions on Intelligent Transportation Systems, 2021

2.1 Introduction

Tremendous progress has been made in estimating human poses "in the wild" driven by popular data collection campaigns [12, 111]. However, low accuracy and high computational complexity have prevented applications to the transportation domain with real-time requirements like self-driving cars and last-mile delivery robots. While a pose estimate is not the final goal, it is an effective low-dimensional and interpretable representation of humans to detect critical actions for autonomous navigation systems. Consequently, the further away a human pose can be detected in real-time, the safer an autonomous system will be. This concept can be extended to other dynamic agents, such as cars at the intersection, or animals crossing the street. We can cast these tasks as detecting and associating semantic keypoints, which are not limited to human poses, *e.g.*, "dog pows" or "left brake lights of vehicles", as illustrated in Figure 2.1a.

This chapter tackles the multi-agent semantic keypoint estimation problem given a single image input. We mainly focus on 2D human poses while also showing performances on cars and animals keypoints. We specifically address challenges that arise in autonomous navigation settings, as shown in Figure 2.1b: (i) wide viewing angle with limited resolution on humans, *i.e.*, a height of 30-90 pixels, and (ii) high-density crowds where pedestrians occlude each other. Our method must also be fast enough to be viable for self-driving cars and robust to real-world variations like lighting, weather and occlusions.

Although pose estimation has been studied before the deep learning era, a significant cornerstone is the work of OpenPose [34], followed by Mask R-CNN [72]. The former is a bottom-up approach (detecting joints without a person detector), and the latter is a top-down one (using a person detector first and outputting joints within the detected bounding boxes). While the performance of these methods is stunning on high enough resolution images, they perform poorly in the limited resolution regime and in dense crowds where humans partially occlude each other.

In this chapter, we propose to extend the notion of fields in pose estimation proposed in [34] to go beyond scalar and vector fields to composite fields. We introduce a new neural network architecture with two head networks: one head network predicts a confidence and precise location of a body part or joint, which we call a Composite Intensity Field (CIF) and which is similar to the fused part confidence map in [138], and the other head network predicts associations between parts, called the Composite Association Field (CAF), which is of a new composite structure. Our encoding scheme has the capacity to store fine-grained information on low resolution activation maps.

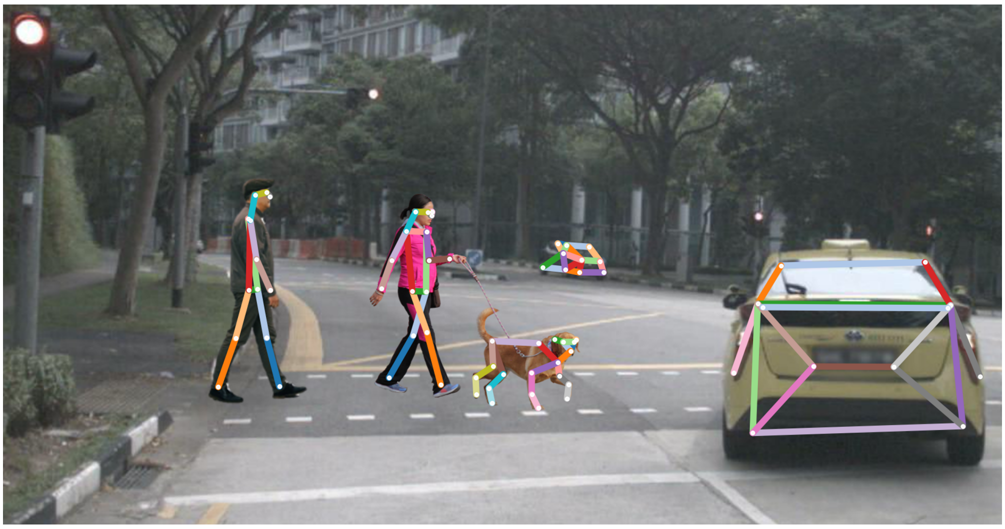
Our experiments show that we outperform all previous methods in accuracy and speed on the CrowdPose dataset [105] with its particularly crowded images. Our method also achieves state-of-the-art results among bottom-up methods on the COCO [111] keypoint task in precision while being an order of magnitude faster in speed. Finally, we show that our method generalizes to car and animal poses, demonstrating its suitability for a holistic perception framework. Our method is implemented as an open source library, referred to as *OpenPifPaf*[I].

2.2 Related Work

2.2.1 Human Semantic Keypoints

The task of detecting human semantic keypoints - also called human pose estimation - has been recently studied using Convolutional Neural Networks [188, 72, 35, 135, 138, 206, 186, 199, 136, 90, 42]. All approaches for human pose estimation can be grouped into bottom-up and top-down methods. The former estimates each body joint first and then groups them to form a unique pose. The latter runs a person detector first and estimates body joints within the detected bounding boxes. Bottom-up methods were pioneered, *e.g.*, by Pishchulin *et al.* with DeepCut [145]. In their work, the part association is solved with an integer linear program leading to processing times for one image of the order of hours. Newer methods use greedy decoders in combination with additional tools to reduce prediction time as in Part Affinity Fields [35], Associative Embedding [135], PersonLab [138] and multi-resolution networks with associate embedding [42].

[I]https://github.com/openpifpaf/openpifpaf



(a) A real-world scene from the perspective of a self-driving car. Schematically, all moving actors are detected with their semantic keypoints. We place particular emphasis on understanding humans but also show generalizations to animals and cars. Here, a car is running a red light while also swerving to the right to avoid a woman who is walking her dog.

(b) A very crowded scene in the city of Lausanne. Our algorithm estimates semantic keypoints for more than 800 people.

Figure 2.1 – We want to estimate human semantic keypoints in the transportation domain from a self-driving car perspective (a), and in a crowded scene where humans occupy small portions of the image (b).

2.2.2 Beyond Humans

While many state-of-the-art methods focused on human body pose detection, the research community has recently studied their performance on other classes such as animals and cars.

Pose estimation research for animals and cars has to deal with additional challenges: limited labeled data [33] and large number of self-occlusions [159].

For animals, datasets are usually small and include limited animal species [142, 33, 26, 108, 124]. To overcome this issue, DeepLabCut [123] and WS-CDA [33] have developed transfer learning techniques from humans to animals. Mu *et al.* [129] have generated a synthetic dataset from CAD animal models and proposed a technique to bridge the real-synthetic domain gap. Another line of work has extended the human SMPL model [113] to animals to learn simultaneously pose and shape of endangered animals [221, 27, 220].

For cars, self-occlusions between keypoints are inevitable. A few methods improve performances by estimating 2D and 3D keypoints of vehicles together. Occlusion-net [159] uses a 3D graph network with self-supervision to predict 2D and 3D keypoints of vehicles using the CarFusion dataset [51], while GSNet [84] predicts 6DoF car pose and reconstructs dense 3D shape simultaneously. Without 3D information, the popular OpenPose [34] shows qualitative results for vehicles and Simple Baseline [171] extends a top-down pose estimator for cars on a custom dataset based on Pascal3D+ [205].

2.3 Method

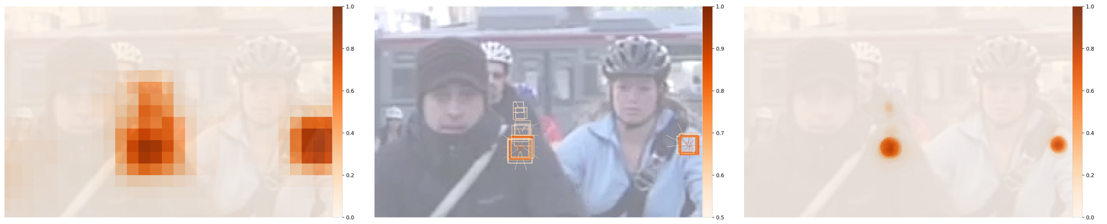
We aim to present a method that can detect and associate semantic keypoints efficiently. We place particular emphasis on urban and crowded scenes that are difficult for autonomous vehicles. Many previous methods struggle when object bounding boxes overlap. In bird-eye views from drones or security cameras, bounding boxes are more separated than in a car driver's perspective. Here, top-down methods struggle. Previous bottom-up methods have been trailing top-down methods in accuracy without improving on performance either. Our bottom-up method is efficient, employs a stable field representation and has high accuracy and performance that even surpasses top-down methods. Figure 2.5 presents our model architecture. It is a shared ResNet [73] or ShuffleNetV2 [118] base network without max-pooling.

2.3.1 Composite Fields

Our method relies on the *Composite Fields* formalism to detect semantic keypoints. Hereafter, we briefly present them.

Field Notation. Fields are functions over locations (*e.g.*, feature map cells) and their outputs are primitives like scalars or composites. Composite Fields jointly predict multiple variables of interest, for example, the confidence, precise location and size of a semantic keypoint (*e.g.*, body joint).

We will enumerate the spatial output coordinates of the neural network with i, j and reserve x, y



(a) (b) (c)

Figure 2.2 – Visualizing the components of the CIF for the "left shoulder" keypoint on a small image crop. The confidence map is shown in (2.2a). The vector field with joint-scale estimates is shown in (2.2b). Only locations with confidence > 0.5 are drawn. The fused confidence, vector and scale components according to Equation 2.1 are shown in (2.2c).

(a) (b)

Figure 2.3 – Visualizing the components of the CAF that associates left shoulder with left hip. This is one of the 18 CAF. Every location of the feature map is the origin of two vectors which point to the shoulders and hips to associate. The confidence of associations $\mathbf{a}_c$ is shown at their origin in (2.3a) and the vector components for $\mathbf{a}_c$ greater than 0.5 are shown in (2.3b).

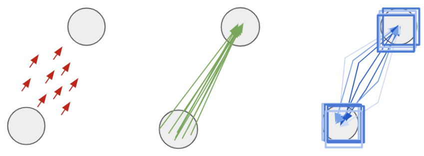


Figure 2.4 – Common association fields between two joints. Joints are visualized as gray circles. Part Affinity Fields (a) as used in OpenPose [35] are unit vectors indicating a direction towards the next joint. Mid-range fields (b) as used in PersonLab [138] are vectors originating in the vicinity of a source joint and point to the target joint. Our Composite Association Field (c) regresses both source and target points and additionally predicts their joint size which are visualized with blue squares.

for real-valued coordinates in the input image. A field over (i, j) is denoted with $\mathbf{f}^{ij}$ and can have scalar, vector or composite values. For example, the composite field of scalars s and 2D vector components v_x, v_y is $\{s, v_x, v_y\}^{ij}$. This is equivalent to "overlaying" a confidence map with a vector field if the ground truth is aligned. This equivalence is trivial in this example but becomes more subtle when we discuss association fields below.

Composite Intensity Fields (CIF). The Composite Intensity Fields (CIF) characterize the intensity of semantic keypoints. The composite structure is based on [139] with the extension of a scale σ to characterize the keypoint size. This is identical to the part intensity field in [95]. We use the notation $\mathbf{p}_J^{ij} = \{c, x, y, b, \sigma\}_J^{ij}$ where J is a particular body joint type, c is the confidence, x and y are regressed coordinates, b is the uncertainty in the location and σ is the size of the joint.

Figure 2.2 shows the components of a CIF field and a high resolution accumulation of the predicted intensity. The field is coarse with a stride of 16 with respect to the input image but the accumulated intensity is at high resolution. The high resolution confidence map $f(v, w)$ is a convolution of an unnormalized Gaussian kernel $\mathcal{N}$ with width σ over the regressed targets from the Composite Intensity Field x and y weighted by its confidence c:

$$f_J(v, w) = \sum_{ij} c_J^{ij} \, \mathcal{N}(v, w | x_J^{ij}, y_J^{ij}, \sigma_J^{ij}) \tag{2.1}$$

where v and w are real-valued coordinates in the image. This accumulation incorporates information of the confidence c, the precisely regressed spatial location (x, y) and the predicted joint size σ. This map f_J is used to seed the pose decoder and to rescore predicted CAF associations.

Composite Association Fields (CAF). Efficiently forming associations is the core challenge for differentiating between people. The most difficult cases are crowded scenes and camera angles where people occlude other people – as is the case in the self-driving car perspective where pedestrians occlude other pedestrians. Top-down methods first estimate bounding boxes and then

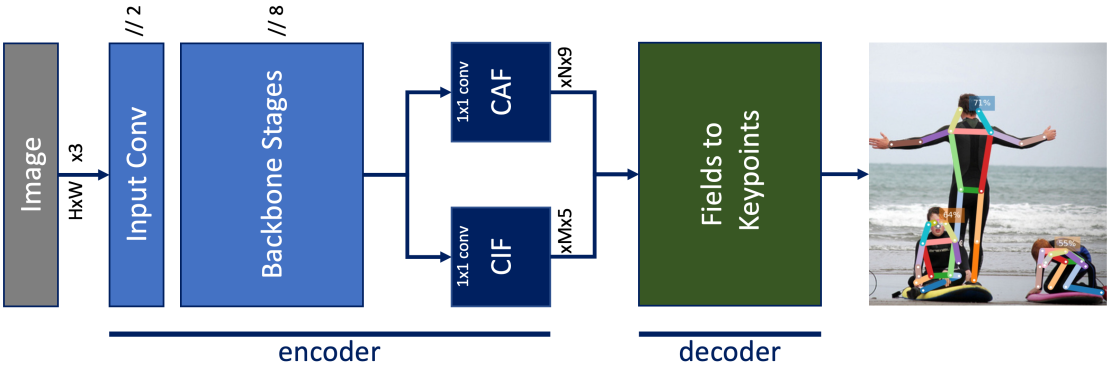


Figure 2.5 – Model architecture. The input is an image batch of size (H,W) with three color channels, indicated by "x3". The neural network based encoder produces composite fields for M joints and N connections. An operation with stride two is indicated by "//2". The shared backbone is a ResNet [73] or ShuffleNetV2 [118] without max-pooling. We use a single 1×1 convolution in each head network, and for optional spatial upsampling, we append a sub-pixel convolution layer [176] to each head network. The decoder converts a set of composite fields into pose estimates. Each semantic keypoint is represented by a confidence score, a real-valued (x, y) coordinate pair and a size estimate.

do single-person pose estimation per bounding box. This assumes non-overlapping bounding boxes which is not given in our scenario. Therefore, we focus on bottom-up methods.

We introduce Composite Association Field (CAF) to connect joint locations together into poses. Our association fields differ from OpenPose's Part Affinity Fields [35] and PersonLab's mid-range fields [138]. A graphical review of association fields is shown in Figure 2.4 and shows that our CAF expresses the most detail about an association.

CAFs predict a confidence, two vectors to the two parts this association is connecting, two spreads b for the spatial precisions of the regressions (details in Section 2.3) and two joint sizes σ. CAFs are represented with $\mathbf{a}^{ij}_{J_1 \leftrightarrow J_2} = \{c, x_1, y_1, x_2, y_2, b_1, b_2, \sigma_1, \sigma_2\}^{ij}_{J_1 \leftrightarrow J_2}$ where $J_1 \leftrightarrow J_2$ is the association between body joints J_1 and J_2. Predicted associations between left shoulders and left hips are shown for an example image in Figure 2.3. In our representation of an association, physically meaningful quantities are regressed to continuous variables and do not suffer from the discreteness of the feature map. In addition, it is important to represent associations between two joints that are at the same pixel location. Our representation is stable for these zero-distance associations – something that Part Affinity Fields [35] cannot do.

2.3.2 Loss Functions for Composite Fields

Human pose estimation algorithms tend to struggle with the diversity of scales that a human pose can have in an image. While a localization error for the joint of a large person can be minor, that same absolute error might be a major mistake for a small person. Our loss is the logarithm of the probability that all components are "well" predicted, *i.e.*, it is the sum of the log-probabilities for the individual components. Each component follows standard loss prescriptions. We use binary cross entropy (BCE) for classification with a Focal loss modification w [110]. To regress locations in the image, we use the Laplace loss [85] which is an L_1-type loss that is attenuated by a predicted spread $\hat{b}$ in the location. To regress additional scale components (keypoint sizes), we use a Laplace loss with a fixed spread $b_\sigma = 3$. The CIF loss function is:

$$\mathscr{L}_{\text{CIF}} = \sum_{m_c} w(c,\hat{c}) \text{BCE}(c,\hat{c}) \tag{2.2}$$

$$+ \sum_{m_v} \frac{1}{\hat{b}} L_2(v,\hat{v},b_{\min}) + \log \hat{b} \tag{2.3}$$

$$+ \sum_{m_s} \frac{1}{b_s} \left| 1 - \frac{\hat{s}}{s} \right| \tag{2.4}$$

with its three parts for confidence (2.2), localization (2.3) and scale (2.4) and where:

$$L_2(v,\hat{v},b_{\min}) = \sqrt{(v_1-\hat{v}_1)^2 + (v_2-\hat{v}_2)^2 + b_{\min}^2} \; . \tag{2.5}$$

The sums are over masked feature cells m_c, m_v and m_σ with i, j, J implied. The mask for confidence m_c is almost the entire image apart from regions annotated as "crowd regions" [111]. The masks for localization m_v and for scale m_σ are only active in a 4×4 window around the ground truth keypoint. Per feature map cell, there is a ground truth confidence c and its predicted counterpart $\hat{c}$. The predicted location $\hat{v} = (\hat{v}_1, \hat{v}_2)$ is optimized with a Laplace loss with a predicted spread $\hat{b}$ for heteroscedastic aleatoric uncertainty [85] with respect to the ground truth location v. A $b_{\min} = 1$px is added to prevent exploding losses when the spread becomes too small. For stability, we clip the BCE loss when it becomes larger than five. The CAF loss has the same structure but with two localization components (2.3) and two scale components (2.4).

Self-Hidden Keypoint Suppression. The COCO evaluation metric treats visible and hidden keypoints in the same manner. As in [95], we include hidden keypoints in our training. However, when a visible and a hidden keypoint appear close together, we remove the hidden keypoint from the ground truth annotation so that this keypoint is not included in associations. In Figure 2.6, we show the effect of excluding these self-hidden keypoints from training and observe better pose reconstruction when a keypoint hides another keypoint of the same type.

2.3.3 Greedy Decoder

The composite fields are converted into sets of pose estimates with a greedy decoder. The CIF field and its high-resolution accumulation $f(x,y)$ defined in equation 2.1 provide seed locations. Previously, new associations were formed starting at the joint that has currently the highest score without taking the CAF confidence of the association into account. Here, we introduce a frontier

Figure 2.6 – Effect of self-hidden keypoint suppression during training. The left image is without and the right image is with self-hidden keypoint suppression. The left hips of both soccer players collide in pixel space.

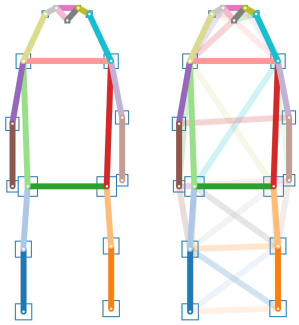
which is a priority queue of possible next associations. The frontier is ordered by the possible future joint scores which are a function of the previous joint score and the best CAF association:

$$\max_{ij} s(\mathbf{a}^{ij}_{J_1 \leftrightarrow J_2}, \vec{x}) = c \exp\left(-\frac{||\vec{x} - (x_1, y_1)||_2}{\sigma_1}\right) f_{J_2}(x_2, y_2) \tag{2.6}$$

where $\vec{x}$ is the source joint location, $\mathbf{a}^{ij}_{J_1 \leftrightarrow J_2} = (c, x_1, y_1, x_2, y_2, \sigma_1, \sigma_2)$ is the CAF field with implied sub-/superscripts on the components and f_{J_2} is the high resolution confidence map of the target joint J_2. An association is rejected when it fails reverse matching. To reduce jitter, we not only use the best CAF association in the above equation but a weighted mixture of the best two associations; similar to blended connections in [19]. Only when all possible associations are added to the frontier, the connection is made to the highest priority in the frontier. This algorithm is fast and greedy. Once a connection to a new joint has been made, this decision is final.

Instance Score and Non-Maximum Suppression (NMS). Once all poses are reconstructed, we apply NMS. Poses are first sorted by their instance score which is the weighted mean of the keypoint scores where the three highest keypoint scores are weighted three times higher. We run NMS at the keypoint level as in [95, 138]. The suppression radius is dynamic and based on the predicted joint size. We do not refine predictions.

Denser Pose Skeletons. Figure 2.7 gives an overview of the pose skeletons that are used in this paper. In particular, Figure 2.7b shows a modification of the standard COCO pose [111] with additional associations. These denser associations are redundancies in case of occlusions. The additional associations are longer-range and therefore harder to predict. The frontier in



(a) (b)

Figure 2.7 – A COCO person pose [111] is shown in (a). Additional denser connections are shown in lighter colors in (b).

Figure 2.8 – **Left:** A sparse pose cannot connect the right arm to the facial keypoints leading to the detection of two separate person instances highlighted by the two white bounding boxes. **Right:** An additional dense connection between the nose and right shoulder leads to a correctly identified single pose.

our greedy decoder takes this difficulty into account and automatically prefers easier, confident associations when available. Qualitatively, the advantage of dense associations is shown in Figure 2.8. With the standard COCO skeleton, the single person's pose skeleton would be divided into two disconnected parts (left image) as indicated by the two white bounding boxes. With the additional denser associations, a single pose is formed (right image).

2.4 Experiments

Self-driving cars must perceive and predict pedestrians and other traffic participants robustly. One of the most challenging scenarios are crowded places. We will first show experiments on human pose estimation in CrowdPose [105] which contains particularly challenging scenarios and on the standardized and competitive COCO [111] person keypoint benchmark. To demonstrate the universality of our approach, we apply our method also to poses of cars and animals.

2.4.1 Datasets

CrowdPose. In [105], the CrowdPose dataset is proposed. It is a selection of images from other datasets with a particular emphasis on how crowded the images are. The crowd-index of an image represents the amount of overlap between person bounding boxes. The authors place particular emphasis on a uniform distribution of the crowd-index in all data partitions. Because this dataset is a composition of other datasets and to avoid contamination, our CrowdPose models are pretrained on ImageNet [47] and then trained on CrowdPose only. The dataset comes with a split of 10,000 images for training, 2,000 for validation and 8,000 images for the test set.

COCO. The de-facto standard for person keypoint prediction is the competitive COCO keypoint task [111]. The test set is private and powers an active leaderboard via a protected challenge server. COCO contains 56,599 diverse training images with person keypoint annotations. The validation and test-dev sets contain 5,000 and 20,288 images.

ApolloCar3D. We generalize our approach to vehicle keypoints using the ApolloCar3D dataset [182], which contains 5,277 driving images at a resolution of 4K and over 60K car instances. The authors defined 66 semantic keypoints in the dataset and, for each car, they provided annotations for the visible ones. For clarity, we choose a subset of 24 semantic keypoints and show quantitative and qualitative results on this dataset.

Animal Dataset. We evaluate the performances of our algorithm on the Animal-Pose Dataset [33], which provides annotations for five categories of animals: dog, cat, cow, horse, sheep for a total of 20 keypoints. The dataset includes 5,517 instances in more than 3,000 images. The majority of these images originally belong to the VOC dataset [54].

2.4.2 Evaluation

Both CrowdPose and COCO follow COCO's keypoint evaluation method. The object keypoint similarity (OKS) score [111] is used to assign a bounding box to each keypoint as a function of the person instance bounding box area. Similar to detection, the metric computes overlaps between ground truth and predicted bounding boxes to compute the standard detection metrics average precision (AP) and average recall (AR).

CrowdPose breaks down the test set at the image level into easy, medium and hard. The easy set contains images with a crowd index in $[0, 0.1]$, the medium set in $[0.1, 0.8]$ and the hard set in $[0.8, 1.0]$. Given the uniform crowd-index distribution, most images of the test set are in the medium category.

COCO breaks down the precision scores at the instance level for medium instances with a

Figure 2.9 – Illustration of OpenPifPaf predictions from the CrowdPose [105] val set with crowd-index *hard* on a sports scene, a family photo and a street scene.

bounding box area of (32 px)^2 to (96 px)^2 and for large instances with a bounding box area larger than (96 px)^2. For each image, pose estimators have to provide the 17 keypoint locations per pose and a total score for each pose. Only the top 20 scoring poses per image are considered for evaluation.

2.4.3 Implementation Details

Neural Network Configuration. All our models are based on ResNet [73] or ShuffleNetV2 [118] base networks and multiple head networks. The base networks have their input max-pooling operation removed as it destroys spatial information. The stride from input image to output feature map is 16 with 2048 features at each location. We apply no additional modifications to the standard ResNet models. We use the standard building blocks of ShuffleNetV2 backbones to construct our custom configurations which we denote ShuffleNetV2K16/K30. A ShuffleNetV2K16 model has the prediction accuracy of a ResNet50 with fewer parameters than a ResNet18. The configuration is specified by the number of output features of the five stages and the number of repetitions of the blocks in each stage. Our ShuffleNetV2K16 has output features (block repeats) of 24 (1), 348 (4), 696 (8), 1392 (4), 1392 (1) and our ShuffleNetV2K30 has 32 (1), 512 (8), 1024 (16), 2048 (6), 2048 (1). Spatial 3×3 convolutions are replaced with 5×5 convolutions which introduces only a small increase in the number of parameters because all spatial convolutions are depth-wise.

Each head network is a single 1×1 convolution followed by a sub-pixel convolution [176] to double the spatial resolution bringing the total stride down to eight. Therefore, the spatial feature map size for an input image of 801px $\times$ 801px is 101×101. The confidence component of a field is normalized with a sigmoid non-linearity and the scale components for joint-sizes are enforced to be positive with a softplus [53].

Augmentations. We apply the standard augmentations of random horizontal flipping, random rescaling with a rescaling factor $r \in [0.5, 2.0]$, random cropping and padding to 385×385 followed by color jittering with 40% variation in brightness and saturation and 10% variation in hue. We also convert a random 1% of the images to grayscale and generate strong JPEG compression artifacts in 10% of the images.

Training Procedure. For ResNet [73] backbones, we use ImageNet [47] pretrained models. ShuffleNetV2 [118] models are trained from random initializations. We use the SGD [29] optimizer with Nesterov momentum [134] of 0.95, batch size of 32 and weight decay of 10^{-5}. The learning rate is exponentially warmed up for one epoch from 10^{-3} of its target value. At certain epochs (specified below), the learning rate is exponentially decayed over 10 epochs by a factor of 10. We employ model averaging [146, 168] to extract stable models for validation. At each optimization step, we update an exponentially weighted version of the model parameters with a decay constant of 10^{-2}.

On CrowdPose, which is a smaller dataset than COCO, we train for 300 epochs. We set the target learning rate to 10^{-5} and decay at epochs 250 and 280.

On COCO, we use a target learning rate of 10^{-4} and decay at epoch 130 and 140. The training time for 150 epochs of a ShuffleNetV2K16 on two V100 is approximately 37 hours. We do not use any additional datasets beyond the COCO keypoint annotations.

2.4.4 Results

Crowded Human Pose Estimation. In Figure 2.9, we show example pose predictions from the CrowdPose [105] validation set. We show results in a diverse selection of sports disciplines

Table 2.1 – Evaluation on the CrowdPose test dataset [105]. Our OpenPifPaf result is based on a ResNet50 backbone with single-scale evaluation at 641px. *Values extracted from CrowdPose paper [105]. +Employs multi-scale testing.

	AP	$AP^{0.50}$	$AP^{0.75}$	AP_{easy}	AP_{medium}	AP_{hard}	FPS
Mask R-CNN* [72]	57.2	83.5	60.3	69.4	57.9	45.8	2.9
AlphaPose* [55]	61.0	81.3	66.0	71.2	61.4	51.1	10.9
HigherHRNet-W48 [42]	65.9	86.4	70.6	73.3	66.5	57.9	-
SPPE [105]	66.0	84.2	71.5	75.5	66.3	57.4	10.1
HigherHRNet-W48+ [42]	67.6	87.4	72.6	75.8	68.1	58.9	-
OpenPifPaf (ours)	**70.5**	**89.1**	**76.1**	**78.4**	**72.1**	**63.8**	**13.7**

Table 2.2 – Evaluation metrics for the COCO 2017 test-dev dataset for bottom-up methods. Numbers are extracted from the respective papers. Our prediction time is determined on a single V100 GPU. *Only evaluating images with three person instances.

	AP	AP^{M}	AP^{L}	t [ms]
OpenPose [35]	61.8	57.1	68.2	100
Assoc. Emb. [135]	65.5	60.6	72.6	166
PersonLab [138]	68.7	64.1	75.5	-
MultiPoseNet [90]	69.6	65.0	76.3	43*
HigherHRNet [42]	70.5	66.6	75.8	>1000
OpenPifPaf (ours)	**71.9**	**68.5**	**77.4**	69

and everyday settings. All shown images are from the *hard* subset with a crowd-index larger than 0.8.

In Table 2.1, we show a quantitative comparison of our performance with other methods. We are not only more precise across all precision metrics AP, $AP^{0.50}$, $AP^{0.75}$, AP_{easy}, AP_{medium} and AP_{hard} but also predict faster than all previous top-performing methods at 13.7 FPS (frames-per-second) on a single GTX1080Ti.

COCO. All state-of-the-art methods compare their performance on the well-established COCO keypoint task [111]. Our quantitative results on the private 2017 test-dev set are shown in Table 2.2 along with other bottom-up methods. This comparison includes field-based methods [35, 138, 95] and methods based on associative embedding [135, 42] and shows that we outperform all the other bottom-up methods. We evaluate on rescaled images where the longer edge is 801 px. We evaluate a single forward pass without horizontal flipping and without multi-scale evaluation because we aim for a fast method. The average time per image with a GTX1080Ti is 152 ms (63 ms on a V100) of which 29 ms is used for decoding.

Pedestrian, Car and Animal Poses. A holistic perception framework for autonomous vehicles also needs to be able to generalize to other classes than humans. We show that we can predict

Figure 2.10 – Qualitative results from the KITTI [64] and ApolloCar3D [182] datasets. We resolve distant pedestrians, cyclists and cars and handle changing lighting conditions well.

Figure 2.11 – Qualitative results from the Animal-Pose dataset [33]. The left image was processed by a person model and an animal model.

Table 2.3 – Quantifying detection performance for pedestrians, cars and animals. In the "Pedestrians" column, we show the detection rate on KITTI [64] with IoU=0.3 and instance threshold of 0.2 for all methods. For "Vehicles", we show the keypoint detection rate on ApolloCar3D [182] which was published in previous methods and we also provide AP in the text. In the "Animals" column, we provide keypoint AP as defined in the Animal-Pose dataset [33].

Method	Pedestrians	Vehicles	Animals
Mono3D [41]	73.2	-	-
3DOP (stereo) [40]	73.1	-	-
MonoDIS [180]	60.5	-	-
SMOKE [112]	39.1	-	-
MonoPSR [99]	82.8	-	-
CPM [199]	-	75.4	-
WS-CDA [33]	-	-	44.3
OpenPifPaf (ours)	**84.6**	**86.1**	**47.8**
Human labelers [182]	-	92.4	-

Table 2.4 – Ablation studies of skeleton choice and decoder configurations for human pose estimation. All results (except where explicitly stated otherwise) are produced with the same ShuffleNetV2k16 model on the COCO val set [111] on a single GTX1080Ti. First, we review different backbone architectures (a ResNet50 [73] and a larger ShuffleNetV2 [118]). Second, we show that only using confident keypoints leads to a large drop in precision. Third, we observe that the Frontier decoder is more important for denser skeletons while incurring almost no overhead on sparse skeletons. Fourth, we can produce a memory-efficient version of our decoder at a cost of 1.4% in AP. The biggest drop in accuracy comes from not rescoring the CAF field and the largest contributor to increasing the inference time is not rescoring the seeds.

		AP	$AP^{0.50}$	$AP^{0.75}$	AP^{M}	AP^{L}	t [ms]	t_{dec} [ms]
	original (ShuffleNetV2K16)	**66.8**	**86.5**	**73.2**	**62.1**	**74.6**	**50**	**19**
Backbone	ResNet50	68.2	87.9	74.6	65.8	72.7	64	22
	ShuffleNetV2K30	71.0	88.8	77.7	66.6	78.5	92	16
Keypoints	independent-only	-8.1	-6.3	-9.5	-8.7	-7.3	±0	±0
Decoder	no-frontier	±0.0	-0.1	+0.1	±0.0	-0.1	-1	-1
	dense	+0.1	+0.2	+0.2	-0.3	+0.5	+15	+15
	no-frontier and dense	-0.3	+0.1	-0.1	-0.5	±0.0	+14	+14
Efficiency	no seed rescoring	-0.1	-0.4	-0.1	+0.2	+0.1	+71	+54
	no seed rescoring (with NMS)	+0.1	+0.1	±0.0	+0.2	+0.0	+19	+15
	no CAF rescoring	-1.0	-0.3	-1.0	-1.0	-1.7	-1	-1
	no rescoring, (with NMS)	-1.4	-0.4	-1.4	-1.0	-2.3	+9	+7

poses of cars and animals with high accuracy in Figures 2.10 and 2.11 and provide a quantitative summary in Table 2.3.

The AP metric follows the same protocol of human instances, but to the best of our knowledge no previous method has evaluated AP on ApolloCar3D [182] without leveraging 3D information. Hence, we include a study on the keypoint detection rate, which has been defined in the ApolloCar3D dataset [182] and considers a keypoint to be correctly estimated if the error is less than 10 pixels. Our method achieves a detection rate of 86.1% compared to 75.4% of CPM [199]. Notably, the authors of ApolloCar3D [182] also report the detection rate of the human labelers to be 92.4%.

On animal instances, our model achieves an AP of 47.8%, compared to 44.3% of WS-CDA, the baseline developed by the authors of the Animal-Pose dataset [33]. Lower performances on animals are due to the smaller dataset size with just 4K training instances. Simultaneous training for humans and animals to achieve better generalization is left for future work.

2.4.5 Ablation Studies

We study the impact of the backbone, the precise criteria for a keypoint, our proposed Frontier decoder, a memory efficient decoder, and the impact of input image size. Our studies on the COCO val set are run with an option to force complete poses. This is the common practice as the

COCO metric does not penalize false positive keypoints within poses. This option would not be used in most real-world settings. Without forcing complete poses, the decoding time and the total prediction time are reduced by about 10ms.

Backbone. The reference backbone is a small ShuffleNetV2K16. We show comparisons to the larger ResNet50 and ShuffleNetV2K30 backbones and show how they improve precision (AP) and at what cost in timing.

Keypoint Criterium. We try to illuminate why our precision and speed is significantly better than OpenPose [35]. OpenPose first detects keypoints and then associates them. Therefore, every keypoint has to be detectable individually. In our method, on the other side, new keypoint associations are generated from a source keypoint. These new keypoints are not previously known, they are discovered in the association. That allows OpenPifPaf to generate poses from a strong seed keypoint and connect to less confident keypoints. In "independent-only", we restrict the keypoints of OpenPifPaf to be all of the quality of an independent seed keypoint and observe a dramatic drop of 8.1% in AP.

Frontier Decoder. Next, we study the impact of the Frontier decoder with respect to a simpler decoder without frontier. The standard pose is sparsely connected and, therefore, the frontier only has few alternatives to prioritize. For a denser pose ("dense"), the impact of the frontier (compare with "no-frontier and dense") is more pronounced (+0.3 AP).

Memory Efficient Decoding. In the bottom part of Table 2.4, we study the effect of removing the high-resolution accumulation map (HR) to reduce the memory footprint. This high resolution map is used in two places. First, to rescore the seeds and, second, to rescore the CAF. The impact of the seed rescoring is only 0.1 in AP but comes at a large cost in decoding time. As an alternative, we investigate a local non-maximum suppression (NMS) that selects a seed only if it is the highest confidence in a 3×3 window (introduced in CenterNet [218]). This NMS reduces the decoding time but not back to the original speed. Independently, we study the impact of rescoring the CAF field which is about +1.0% in AP. Only when both the seed rescoring and the CAF rescoring are removed, the creation of the HR maps can be omitted. In that memory efficient configuration (bottom line in Table 2.4), the AP dropped by 1.4% with respect to "original". This demonstrates the importance of the high-resolution accumulation for speed and accuracy and which should only be removed when absolutely necessary.

2.5 Conclusion

We have demonstrated a new method for bottom-up pose detection for 2D human poses and shown its strength in crowded and occluded scenes relevant for perception in self-driving cars and social robots. We outperform previous state-of-the-art methods on CrowdPose and COCO datasets while running an order of magnitude faster. We have also shown that our method generalizes to pose estimation of cars and animals, emphasizing how semantic keypoints can represent any object. In the rest of this thesis, we will demonstrate that semantic keypoints are an excellent intermediate representation to 3D locate humans and estimate their actions in the context of autonomous driving applications.

3 Monocular 3D Pedestrian Localization and Uncertainty Estimation

This chapter is based on the articles:

Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *MonoLoco: Monocular 3D Pedestrian Localization and Uncertainty Estimation*, The IEEE International Conference on Computer Vision (ICCV), 2019

Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *Perceiving Humans: From Monocular 3D Localization to Social Distancing*, IEEE Transactions on Intelligent Transportation Systems, 2021

Additional qualitative results are displayed on YouTube[I].

3.1 Introduction

The previous chapter described how to extract semantic keypoints from a monocular image. This chapter uses semantic keypoints as an intermediate representation to improve 3D pedestrian localization for autonomous driving applications.

Autonomous vehicles commonly rely on LiDAR sensing solutions despite high cost and sparsity of point clouds over long ranges [41, 219, 149]. Cost-effective perception systems have been proposed by adopting stereo/multiple cameras to address the fundamental ambiguity of monocular solutions [40, 106]. Yet researchers are studying how to push the limits of monocular perception to further contribute to multi-sensor fusion [109, 125]. In monocular settings, progress has been made estimating 3D positions of vehicles from monocular images [39, 128, 163], while pedestrians have received far less attention due to lack of adequate performances. In fact, inferring 3D locations of pedestrians from a single image is particularly ambiguous due to the variance in human heights and shapes.

[I] https://www.youtube.com/watch?v=ii0fqerQrec

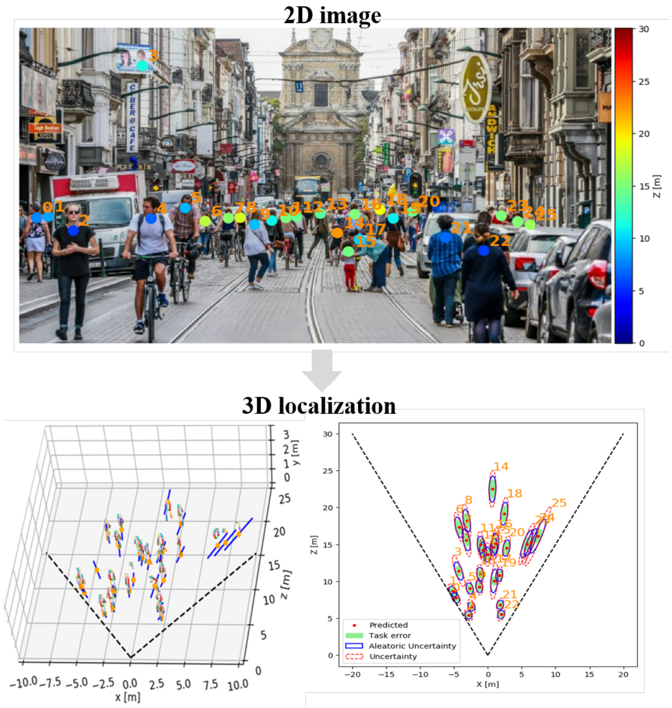


Figure 3.1 – 3D localization of pedestrians from a single RGB image. Our method leverages 2D poses to find 3D locations as well as confidence intervals. The confidence intervals are shown as blue lines in the left 3D view and as ellipses in the right birds-eye-view.

In this chapter, we explicitly study the intrinsic ambiguity of locating pedestrians in the scene and investigate whether we can learn this ambiguity from the data. Driven by this perception task, we aim at providing more insights into the general problem of uncertainty estimation in deep learning.

Kendall and Gal [85] introduced practical uncertainty estimation for deep learning in perception tasks, distinguishing between *aleatoric* and *epistemic* uncertainty [49, 85]. The former models noise inherent in the observations, while the latter is a property of the model parameters and can be reduced by collecting more data. While their proposed measure of uncertainty is inspiring, they could not compare it with a known uncertainty, referred to as a *task error*. In this chapter, based on the statistical variation of human height within the adult population [194], we quantify the ambiguity of the task, *i.e.*, the task error: an upper bound of performances for monocular 3D pedestrian localization. Surprisingly, the task error is reasonably low. Our experiments show accurate results in 3D localization without overcoming the limitation due to this intrinsic ambiguity.

We propose a simple probabilistic method for monocular 3D localization tailored for pedestrians that works with any fixed or moving cameras, and does not rely on ground plane estimation. We specifically address the challenges of the ill-posed task by predicting confidence intervals in contrast to point estimates, which account for aleatoric and epistemic uncertainties. Our method is composed of two distinct steps. First, we leverage our pose detector described in Chapter 2 to obtain 2D keypoints, a low-dimensional meaningful representation of humans. Second, we input the detected keypoints to a light-weight feed-forward network and output the 3D location of each instance along with a confidence interval. We explore whether 2D keypoints contain enough information for a network to learn the intrinsic ambiguity of the task as well as accurate localization. We leverage a recently introduced loss function based on the Laplace distribution [85] to incorporate aleatoric uncertainty for each predicted location without direct supervision at training time. MC dropout at inference time is used to capture epistemic uncertainty [60]. Furthermore, our network, referred to as MonoLoco++, independently learns the distribution of uncertainties, predicting confidence intervals comparable with the corresponding task error. The code[II] and a video[III] with qualitative results are available online.

3.2 Related Work

Monocular 3D Object Detection. Recent approaches for monocular 3D object detection in the transportation domain focused only on vehicles as they are rigid objects with known shape. To the best of our knowledge, only a few methods valuated pedestrians from monocular RGB images. The very recent MonoPSR [99] leveraged point clouds at training time to learn local shapes of objects. MonoDIS [180] proposes to disentangle the contribution of each loss component, while SMOKE [112] combines a single keypoint estimate with regressed 3D variables. Kundegorski and Breckon [100] achieved reasonable performances combining infrared imagery and real-time photogrammetry. Alahi *et al*. combined monocular images with wireless signals [6] or with additional visual priors [4, 5]. The seminal work of Mono3D [39] exploited deep learning to create 3D object proposals for *car*, *pedestrian* and *cyclist* categories but it did not evaluate 3D localization of pedestrians. It assumed a fixed ground plane orthogonal to the camera and the proposals were then scored based on scene priors, such as shape, semantic and instance segmentations. Following methods continued to leverage Convolutional Neural Networks and focused only on *Car* instances. To regress 3D pose parameters from 2D detections, Deep3DBox [128], MonoGRnet [151], and Hu *et al*. [77] used geometrical reasoning for 3D localization, while Multi-fusion [208] and ROI-10D [120] incorporated a module for depth estimation. Recently, Roddick *et al*. [163] escaped the image domain by mapping image-based features into a birds-eye view representation using integral images. Another line of work fits 3D templates of cars to the image [203, 204, 38, 101]. While many of the related methods achieved reasonable performances for vehicles, current literature lacks monocular methods addressing other categories in the context of autonomous driving, such as pedestrians and cyclists.

Uncertainty Estimation in Computer Vision. Deep neural networks need to have the ability not only to provide the correct outputs but also a measure of uncertainty, especially in safety-critical scenarios like autonomous driving. Traditionally, Bayesian Neural Networks [162, 132]

[II] https://github.com/vita-epfl/monoloco
[III] https://www.youtube.com/watch?v=ii0fqerQrec

were used to model epistemic uncertainty through probability distributions over the model parameters. However, these distributions are often intractable and researchers have proposed interesting solutions to perform approximate Bayesian inference to measure uncertainty, including Variational Inference [66, 28, 170] and Deep Ensembles [104]. Alternatively, Gal *et al*. [60, 61] showed that applying dropout [183] at inference time yields a form of variational inference where parameters of the network are modeled as a mixture of multivariate Gaussian distributions with small variances. This technique, called Monte Carlo (MC) dropout, became popular also due to its adaptability to non-probabilistic deep learning frameworks. Very recently, Postels *et al*. [148] proposed a sampling-free method to approximate epistemic uncertainty, treating noise injected in a neural network as errors on the activation values. In computer vision, uncertainty estimation using MC dropout has been applied for depth regression tasks [85, 148], scene segmentation [130, 85] and, more recently, LiDAR 3D object detection for cars [57].

2D Keypoints for 3D Vision Tasks. Detecting people in images and estimating their skeleton is a widely studied problem and in Chapter 2, we have extensively compared top-down and bottom-up approaches and proposed a method tailored for autonomous driving scenarios that performs well in low-resolution, crowded and occluded scenes [95, 94] . The idea of leveraging 2D keypoints for 3D vision tasks is not novel. The most related to our work is Simple Baseline [122], which shows the effectiveness of latent information contained in 2D semantic keypoints. They achieve state-of-the-art results by simply predicting 3D body joints from 2D poses through a light, fully connected network. However, these lines of work estimate relative 3D joint positions [127, 213, 164], or relative 3D meshes [114, 82], not providing any information about the real 3D location in the scene.

3.3 3D Localization Ambiguity

Inferring distance of pedestrians from monocular images is a fundamentally ill-posed problem. The majority of previous works has circumvented this challenge by assuming a planar ground plane and estimating an homography by manual measurement or by knowing some reference elements [172, 45, 39, 40]. These approaches do not work when people are on stairs and require a static calibrated setup. In this chapter, we address this limitation by directly estimating distance of humans without relying on a ground plane, homography, or contextual cues, such as scene geometry. This problem is ill-posed due to human variation of height. If every pedestrian has the same height, there would be no ambiguity. However, does this ambiguity prevent from robust localization? This section is dedicated to explore this question and analyze the maximum accuracy expected from monocular pedestrian localization.

Our approach consists in assuming that all humans have the same height h_{mean} and analyzing the error of this assumption. Inspired by Kundegorski and Breckon [100], we model the localization error due to variation of height as a function of the ground truth distance from the camera, which we call *task error*. From the triangle similarity relation of human heights and distances,

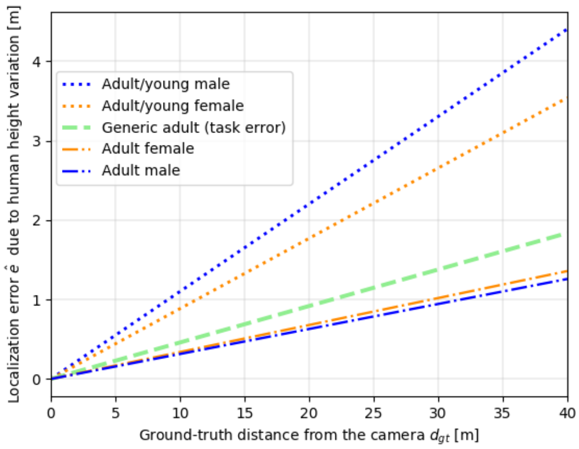


Figure 3.2 – Localization error due to human height variations at different distances from the camera. We approximate the distribution of height for a generic adult as Gaussian mixture distribution and we define the *task error*: an upper bound of performances for monocular methods.

$d_{\text{h-mean}}/h_{\text{mean}} = d_{gt}/h_{gt}$, where h_{gt} and d_{gt} are the the ground-truth human height and distance, h_{mean} is the assumed mean height of a person and $d_{\text{h-mean}}$ the estimated distance under the h_{mean} assumption. We can define the task error for any person instance in the dataset as:

$$e \equiv |d_{gt} - d_{h-mean}| = d_{gt} \left| 1 - \frac{h_{mean}}{h_{gt}} \right| \quad . \tag{3.1}$$

Previous studies from a population of 63,000 European adults have shown that the average height is $178cm$ for males and $165cm$ for females with a standard deviation of around $7cm$ in both cases [194]. However, a pose detector does not distinguish between genders. Assuming that the distribution of human stature follows a Gaussian distribution for male and female populations [59], we define the combined distribution of human heights, a Gaussian mixture distribution $P(H)$, as our unknown ground-truth height distribution. The *expected task error* becomes

$$\hat{e} = d_{gt}\ E_{h \sim P(H)} \left[\left| 1 - \frac{h_{mean}}{h} \right| \right] \quad , \tag{3.2}$$

which represents a lower bound for monocular 3D pedestrian localization due to the intrinsic ambiguity of the task. The analysis can be extended beyond adults. A 14-year old male reaches about 90% of his full height and a female about 95% [59, 100]. Including people down to 14 years old leads to an additional source of height variation of 7.9% and 5.6% for men and women, respectively [100]. Figure 3.2 shows the expected localization error $\hat{e}$ due to height variations in different cases as a function of the ground-truth distance from the camera d_{gt}. This analysis shows that the ill-posed problem of localizing pedestrians, while imposing an intrinsic limit, does not prevent from robust localization in general cases.

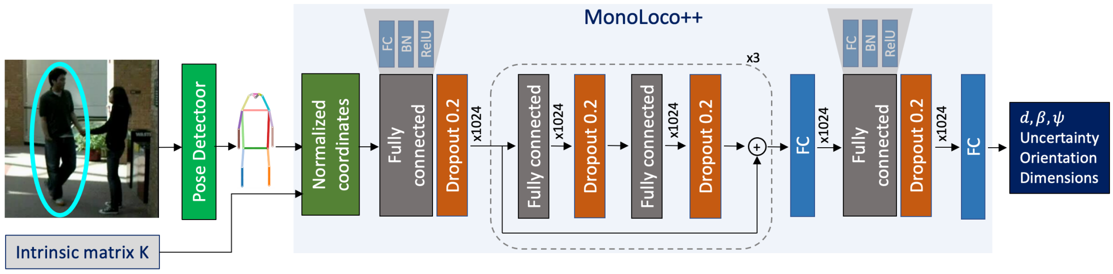


Figure 3.3 – Network architecture. the input is a set of 2D keypoints extracted from a raw image and the output is the 3D location, orientation and dimensions of a human with the localization uncertainty. 3D location is estimated with spherical coordinates: radial distance d, azimuthal angle β, and polar angle ψ. Every fully connected layer (FC) outputs 1024 features and is followed by a Batch Normalization layer (BN) [79] and a ReLU activation function.

3.4 Method

3.4.1 3D Pedestrian Detection

The task of 3D object detection is defined as detecting 3D location of objects along with their orientation and dimensions [64, 31]. In the case of pedestrians, the ambiguity of the task derives from the localization component, which is our main focus. Hence, we argue that effective monocular localization implies not only accurate estimates of the distance but also realistic predictions of uncertainty. Consequently, we propose a method which learns the ambiguity from the data without supervision and predicts confidence intervals in contrast to point estimates. The task error modeled in Eq. 3.2 allows to compare the predicted confidence intervals with the intrinsic ambiguity of the task.

Input. We use a pose estimator to detect a set of keypoints $[u_i, v_i]^T$ for every instance in the image. We then back-project each keypoint i into normalized image coordinates $[x_i^*, y_i^*, 1]^T$ using the camera intrinsic matrix K:

$$[x_i^*, y_i^*, 1]^T = K^{-1} [u_i, v_i, 1]^T \, . \tag{3.3}$$

This transformation is essential to prevent the method from overfitting to a specific camera.

2D Human Poses. We obtain 2D joint locations of pedestrians using the off-the-shelf pose detectors OpenPifPaf [94], a state-of-the-art, bottom-up method designed for crowded scenes and occlusions. The detector can be regarded as a stand-alone module independent from our network, which uses 2D keypoints as inputs. OpenPifPaf has not been fine-tuned on any additional dataset for 3D object detection as no annotations for 2D poses are available.

Output. For each person in the scene, we predict their 3D location using a regressive model, along with their orientation and bounding-box dimensions in a multi-task setting. Estimating depth is arguably the most critical component in vision-based 3D object detection due to intrinsic limitations of monocular settings described in Section 3.3. However, due to perspective projections, an error in depth estimation z would also affect the horizontal and vertical components x and y. To disentangle the depth ambiguity from the other components, we use a spherical coordinate system (r, β, ψ), namely radial distance d, azimuthal angle β, and polar angle ψ. Another advantage of using a spherical coordinate system is that the size of an object projected onto the image plane directly depends on its radial distance d and not on its depth z. The same pedestrian in front of a camera or at the margin of the camera field-of-view will appear as having the same height in the image plane, as long as the distance from the camera d is the same.

As already noted in [106], the viewpoint angle is not equal to the object orientation as people at different locations may share the same orientation θ but results in different projections. Hence, we predict the viewpoint angle α, which is defined as $\alpha = \theta + \beta$, where β denotes the azimuth of the pedestrian with respect to the camera. Similarly to [106], we also parametrize the angle as $[\sin\alpha, \cos\alpha]$ to avoid discontinuity. Regarding bounding box dimensions, we follow the standard procedure to calculate width, height and length of each pedestrian. We calculate average dimensions from the training set and regress the displacement from the expectation.

Our final loss is the logarithm of the probability that all components are "well" predicted, i.e., it is the sum of the log-probabilities for the individual components. For every component but the 3D localization, we use a vanilla L1 loss. To regress distances of people, we use a Laplace-based L1 loss [88], which we describe in the following section. Our minimization objective is a simple non-weighted sum of each loss function. Our minimization objective for our multi-output model follows the formulation in [86].

Base Network. The building blocks of our model are shown in Figure 3.3. The architecture, inspired by Martinez *et al.* [122], is a simple, deep, fully-connected network with six linear layers with 1024 output features. It includes dropout [183] after every fully connected layer, batch-normalization [79] and residual connections [73]. The model contains approximately 400k training parameters.

MonoLoco vs MonoLoco++. We have developed two versions of our network: MonoLoco [23] and MonoLoco++ [24]. In this chapter, we refer to the most up-to-date version (MonoLoco++), and in the experiments section, we compare both versions of our algorithm. Technically, MonoLoco++ differs from MonoLoco [23] by:

- the multi-task approach to combine 3D localization, orientation and bounding-box dimensions,
- the use of spherical coordinates to disentangle the ambiguity in the 3D localization task,

- an improved neural network architecture.

3.4.2 Uncertainty

In this chapter, we propose a probabilistic network which models two types of uncertainty: *aleatoric* and *epistemic* [49, 85]. Aleatoric uncertainty is an intrinsic property of the task and the inputs. It does not decrease when collecting more data. In the context of 3D monocular localization, the intrinsic ambiguity of the task represents a quota of aleatoric uncertainty. In addition, some inputs may be more noisy than others, leading to an input-dependent aleatoric uncertainty. Epistemic uncertainty is a property of the model parameters, and it can be reduced by gathering more data. It is useful to quantify the ignorance of the model about the collected data, *e.g.*, in case of out-of-distribution samples.

Aleatoric Uncertainty. Aleatoric uncertainty is captured through a probability distribution over the model outputs. We define a relative Laplace loss based on the negative log-likelihood of a Laplace distribution as:

$$L_{\text{Laplace}}(x|d,b) = \frac{|1-d/x|}{b} + \log(2b) \quad , \tag{3.4}$$

where x represents the ground-truth distance, and d, b the predicted distance and the spread, making this training objective an attenuated L_1-type loss via spread b. During training, the model has the freedom to ignore noisy data and attenuate its gradients by predicting a large spread b. As a consequence, inputs with high uncertainty have a small effect on the loss, making the network more robust to noisy data. The uncertainty is estimated in an unsupervised way, since no supervision is provided. At inference time, the model predicts the distance d and a spread b which indicates its confidence about the predicted distance. The latter one indicates the model's confidence about the predicted distance. Following [85], to avoid the singularity for $b = 0$, we apply a change of variable to predict the log of the spread $s = \log(b)$.

Compared to previous methods [85, 202], we design a Laplace loss which works with relative distances to keep into account the role of distance in our predictions. For example in autonomous driving scenarios, estimating the distance of a pedestrian with an absolute error can lead to a fatal accident if the person is very close, or be negligible if the same human is far away from the camera.

Epistemic Uncertainty. To model epistemic uncertainty, we follow [60, 85] and consider each parameter as a mixture of two multivariate Gaussians with small variances and means 0 and θ. The additional minimization objective for N data points is:

$$L_{\text{dropout}}(\theta, p_{drop}) = \frac{1-p_{drop}}{2N}||\theta||^2 \quad . \tag{3.5}$$

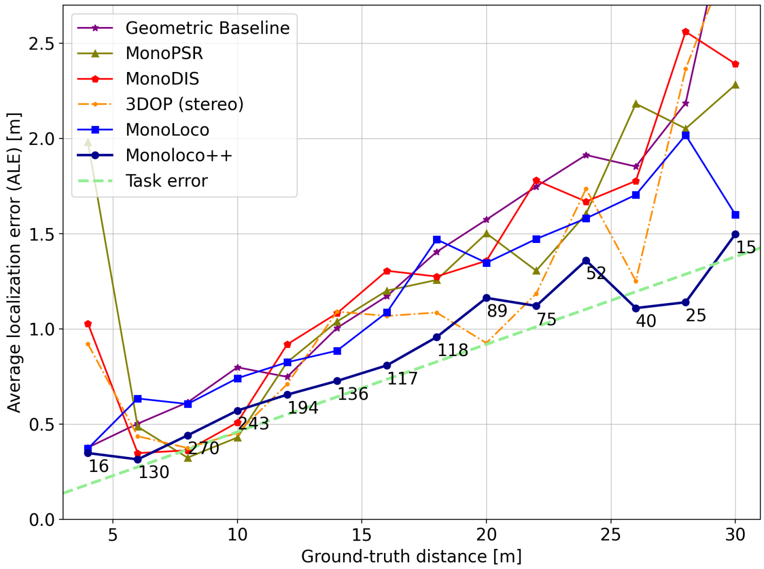


Figure 3.4 – Average localization error (ALE) as a function of distance. We outperform the monocular MonoPSR [99] and MonoDIS [180], while even achieving more stable results than the stereo 3DOP [40]. Monocular performances are bounded by our modeled task error in Eq. 3.2. The task error is only a mathematical construction not used in training and yet it strongly resembles the network error, especially for the more statistically significant clusters (number of predicted instances included).

In practice, we perform dropout variational inference by training the model with dropout before every weight layer and then performing a series of stochastic forward passes at test time using the same dropout probability p_{drop} of training time. The use of fully-connected layers makes the network particularly suitable for this approach, which does not require any substantial modification of the model.

The combined epistemic and aleatoric uncertainties are captured by the sample variance of predicted distances $\tilde{x}$. They are sampled from multiple Laplace distributions parameterized with the predictive distance d and spread b from multiple forward passes with MC dropout:

$$Var(\tilde{X}) = \frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\tilde{x}_{t,i}^{2}(d_t, b_t) - \left[\frac{1}{TI}\sum_{t=1}^{T}\sum_{i=1}^{I}\tilde{x}_{t,i}(d_t, b_t)\right]^2 , \tag{3.6}$$

where for each of the T computationally expensive forward passes, I computationally cheap samples are drawn from the Laplace distribution.

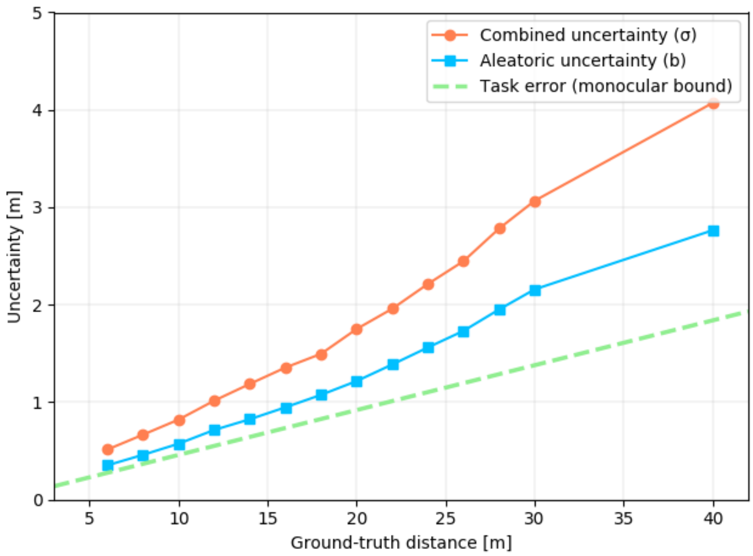


Figure 3.5 – Results of aleatoric uncertainty predicted by MonoLoco++ (spread b), and the modeled aleatoric uncertainty due to human height variation (task error $\hat{e}$). The term $b - \hat{e}$ is indicative of the aleatoric uncertainty due to noisy observations. The combined uncertainty σ accounts for aleatoric and epistemic uncertainty and is obtained applying MC Dropout [60] at test time with 50 forward passes.

3.5 Experiments

3.5.1 Monocular 3D Localization

Datasets. We train and evaluate our monocular model on KITTI Dataset [64]. It contains 7481 training images along with camera calibration files. All the images are captured in the same city from the same camera. To analyze cross-dataset generalization properties, we train another model on the teaser of the recently released nuScenes dataset [31] and we test it on KITTI. We do not perform cross-dataset training.

Training/evaluation Procedure. To obtain input-output pairs of 2D keypoints and distances, we apply an off-the-shelf pose detector and use intersection over union of 0.3 to match our detections with the ground-truths, obtaining 5000 instances for KITTI and 14500 for nuScenes teaser. KITTI images are upscaled by a factor of two to match the minimum dimension of 32 pixels of COCO instances. NuScenes already contains high-definition images, which are not modified. Once the human poses are detected, we apply horizontal flipping to double the instances in the training set.

We follow the KITTI train/val split of Chen *et al.* [39] and we run the training procedure for 200 epochs using Adam optimizer [89], a learning rate of 10^{-3} and mini-batches of 512. The code, available online [II], is developed using PyTorch [141]. Working with a low-dimensional latent representation is very appealing as it allows fast experiments with different architectures and hyperparameters. The entire training procedure requires around two minutes on a single GPU GTX1080Ti.

Evaluation Metrics. We use two metrics to analyze 3D pedestrian localization. First, we consider a prediction as correct if the error between the predicted distance and the ground-truth is smaller than a threshold. We call this metric Average Localization Accuracy (ALA). We use 0.5 meters, 1 and 2 meters as thresholds. We also analyze the average localization error (ALE). To make fair comparison we set the threshold of the methods to obtain similar recall. We do not evaluate on the common set of detected instances; such evaluation would not be reproducible as the common set depends on the methods used for the evaluation. In contrast, analyzing ALE and recall allows for simple but fair comparison. Following KITTI guidelines, we assign to each instance a difficulty regime based on bounding box height, level of occlusion and truncation: *easy*, *moderate* and *hard*. However in practice, each category includes instances from the simpler categories, and, due to the predominant number of easy instances (1240 *easy* pedestrians, 900 *moderate* and 300 *hard* ones), the metric can be misleading and underestimate the impact of challenging instances. Hence, we evaluate each instance as belonging only to one category and add the category *all* to include all the instances.

Geometric Approach. 3D pedestrian localization is an ill-posed task due to human height variations. On the other side, estimating the distance of an object of known dimensions from its projections into the image plane is a well-known deterministic problem. As a baseline, we consider humans as fixed objects with the same height and we investigate the localization accuracy under this assumption.

For every pedestrian, we apply a pose detector to calculate distances in pixels between different body parts in the image domain. Combining this information with the location of the person in the world domain, we analyze the distribution of the real dimensions (in meters) of all the instances in the training set for three segments: head to shoulder, shoulder to hip and hip to ankle. For our calculation we assume a pinhole model of the camera and that all instances stand upright. Using the camera intrinsic matrix K and knowing the ground-truth location of each instance $\mathbf{D} = [x_c, y_c, z_c]^T$ we can back-project each keypoint from the image plane to its 3D location and measure the height of each segment using Eq. 3.3. We calculate the mean and the standard deviation in meters of each of the segments for all the instances in the training set. The standard deviation is used to choose the most stable segment for our calculations. For instance, the position of the head with respect to shoulders may vary a lot for each instance. To take into account noise in the 2D keypoints predictions we also average between left and right keypoints values. The result is a single height Δy_{1-2} which represents the average length of two body parts. In practice, our geometric baseline uses the *shoulder-hip* segment and predicts an average height of 50.5cm. Combining the study on human heights [194] described in Section 3 with the anthropometry study of Drillis *et al*. [52], we can compare our estimated Δy_{1-2} with the human average *shoulder-hip* height: $0.288 * 171.5cm = 49.3cm$.

The next step is to calculate the location of each instance knowing the value in pixels of the chosen keypoints v_1 and v_2 and assuming Δy_{1-2} to be their relative distance in meters. This configuration requires to solve an over-constrained linear system with two specular solutions, of which only one is inside the camera field of view.

Other Baselines. We compare our monocular method on KITTI against four monocular approaches and a stereo one:

- *Mono3D* [39] is a monocular 3D object detector for cars, cyclists and pedestrians. 3D localization of pedestrians is not evaluated but detection results are publicly available
- *MonoPSR* [99] is a monocular 3D object detector that leverages point clouds at training time to learn shapes of objects. In contrast, our method does not use any privileged signal at training time.
- *MonoDIS* [180] is a very recent multi-class 3D object detector that provides evaluations for the pedestrian category on the KITTI dataset.
- *SMOKE* [112] is a single-stage monocular 3D object detection method which is based on projecting 3D points onto the image plane. The authors have shared their quantitative evaluation.
- *3DOP* [40] is a stereo approach for pedestrians, cars and cyclists and their 3D detections are publicly available.

Finally, in Figure 3.4 we also compare the results against the task error of Eq. 3.2, which defines the target error for monocular approaches due to the ambiguity of the task.

3.5.2 Quantitative Results

Table 3.1 summarizes our quantitative results on KITTI. We strongly outperform all the other monocular approaches on all metrics and obtain comparable results with the stereo approach 3DOP [40], which has been trained and evaluated on KITTI and makes use of stereo images during training and test time. In addition, we show cross-dataset generalization properties by training our network on a subset of the nuScenes dataset containing only 1799 instances and evaluating it on the KITTI dataset. Its generalization properties can be attributed to the low-dimensional input space of 2D keypoints [68].

In Figure 3.4, we make an in-depth comparison analyzing the average localization error as a function of the ground-truth distance. We also compare the performances against the *task*

Table 3.1 – Comparing our proposed method against baseline results on the KITTI dataset [64]. We use PifPaf [95] as off-the-shelf network to extract 2D poses. For the ALE metric, we show the recall between brackets to insure fair comparison. We show results by training with three different data splits: KITTI dataset [64], nuScenes teaser [31] or a subset of nuScenes to match the number of instances of the KITTI dataset. All cases share the same evaluation protocol. The models trained on nuScenes show cross-dataset generalization properties by obtaining comparable results in the ALE metric.

Method	Training Dataset	Training Instances	ALE (m) ↓ [Recall (%) ↑]				ALA (%) ↑		
			Easy	*Mod.*	*Hard*	*All*	$< 0.5m$	$< 1m$	$< 2m$
Mono3D [39]	KITTI	1799	2.26 [89%]	3.00 [65%]	3.98 [34%]	2.62 [69%]	13.0	22.9	38.2
MonoPSR [99]	KITTI	1799	0.88 [96%]	1.86 [68%]	1.85 [16%]	1.19 [69%]	31.1	44.2	57.4
SMOKE [112]	KITTI	1799	0.75 [59%]	1.30 [30%]	1.53 [10%]	0.91 [39%]	18.7	27.3	34.5
MonoDIS [180]	KITTI	1799	**0.66** [85%]	1.26 [64%]	1.83 [32%]	0.93 [66%]	33.2	47.6	57.6
3DOP [40] (Stereo)	KITTI	1799	0.67 [88%]	1.19 [64%]	1.93 [37%]	0.94 [69%]	40.6	53.7	61.4
Our Geometric	KITTI	-	1.05 [89%]	0.95 [63%]	1.34 [31%]	1.04 [68%]	23.5	41.9	59.4
Our MonoLoco	KITTI	1799	0.95 [89%]	0.98 [64%]	1.11 [31%]	0.97 [68%]	25.3	43.4	60.5
Our MonoLoco	nuScenes	8189	0.91 [92%]	1.16 [80%]	1.45 [30%]	1.08 [74%]	27.6	46.6	63.7
Our MonoLoco++	KITTI	1799	0.69 [90%]	**0.71** [66%]	1.37 [31%]	**0.76** [70%]	**37.4**	**53.2**	63.6
Our MonoLoco++	nuScenes	1799	0.81 [92%]	0.84 [68%]	1.14 [29%]	0.84 [70%]	31.8	50.2	63.9
Our MonoLoco++	nuScenes	8189	0.72 [91%]	0.77 [68%]	**1.03** [29%]	**0.76** [70%]	32.5	51.9	**65.6**

error due to human height variations modeled in equation 3.2. Our method results in stable performances that almost replicate the target threshold. More generally, it is notable that the error of each method shows a quasi-linear behaviour. At a short range, the majority of methods show large errors, as the instances are not fully visible in the image. Since our method reasons with keypoints, its performances are more stable. At the 25-30m range MonoLoco++ error is slightly lower than the task error. This is mainly caused by the statistical fluctuations due to the small sample sizes at those distances. Figure 3.6 and 3.7 show qualitative results on challenging images from the KITTI and nuScenes datasets, respectively.

Aleatoric Uncertainty. We compare in Figure 3.5 the aleatoric uncertainty predicted by our network through spread b with the *task error* due to human height variation defined in Eq. 3.2. While $\hat{e}$ is a linear function of the distance from the camera, the predicted aleatoric uncertainty (through the spread b) is a property of each set of inputs. In fact, b includes not only the uncertainty due to the ambiguity of the task but also the uncertainty due to noisy observations [85], *i.e.*, the 2D keypoints inferred by the pose detector. Hence, we can approximately define the predictive aleatoric uncertainty due to noisy keypoints as $b - \hat{e}$ and we observe that the further a person is from the camera, the higher is the term $b - \hat{e}$. The spread b is the result of a probabilistic interpretation of the model and the resulting confidence intervals are calibrated. On the KITTI validation set, they include 68% of the instances.

Combined Uncertainty. The combined aleatoric and epistemic uncertainties are captured by sampling from multiple Laplace distributions using MC dropout. During each of the forward passes, we draw and accumulate samples from the estimated Laplace distribution. Then, we

Table 3.2 – Precision and recall of uncertainty for KITTI validation set with 50 stochastic forward passes. $|x-d|$ is the localization error, σ the predicted confidence interval, $\hat{e}$ the task error modeled in Eq. 3.2 and Recall is represented by the % of ground-truth instances inside the predicted confidence interval.

	$\|x-d\|/\sigma$	$\|\sigma-e\|$ [m]	Recall [%]
$p_{drop}=0.05$	0.60	0.90	82.8
$p_{drop}=0.2$	0.58	0.96	84.3
$p_{drop}=0.4$	0.50	1.26	88.3

Table 3.3 – Impact of different loss functions with Mask R-CNN [72] and OpenPifPaf [94] pose detectors on nuScenes teaser validation set [31]. We also show results using the Average Localization Error (ALE) metric as a function of the ground- truth distance using clusters of 10 meters.

Mask R-CNN [72]	ALE [m]				
	$^{10}_{0}$	$^{20}_{10}$	$^{30}_{20}$	$^{+}_{30}$	*All*
Geometric	0.79	1.52	3.17	9.08	3.73
L_1 loss	0.85	**1.17**	2.24	4.11	2.14
Gaussian loss	0.90	1.28	2.34	4.32	2.26
Laplace Loss	**0.74**	**1.17**	2.25	4.12	2.12
OpenPifPaf [94]	ALE [m]				
	$^{10}_{0}$	$^{20}_{10}$	$^{30}_{20}$	$^{+}_{30}$	*All*
Geometric	0.83	1.40	2.15	3.59	2.05
L_1 loss	0.83	1.24	**2.09**	3.32	1.92
Gaussian loss	0.89	1.22	2.14	3.50	1.97
Laplace Loss	0.75	1.19	2.24	**3.25**	**1.90**

calculate the combined uncertainty as the sample variance of predicted distances in Eq. 3.6. The magnitude of the uncertainty depends on the chosen dropout probability p_{drop} in Eq. 3.5. In Table 3.2, we analyze the precision/recall trade-off for different dropout probabilities and choose $p_{\text{drop}} = 0.2$. We perform 50 computationally expensive forward passes and, for each of them, 100 computationally cheap samples from a Laplace distribution using Eq. 3.6. As a result, 84% of pedestrians lie inside the predicted confidence intervals for the validation set of KITTI.

One of our goals is robust 3D estimates for pedestrians, and being able to predict a confidence interval instead of a single regression number is a first step towards this direction. To illustrate the benefits of predicting intervals over point estimates, we construct a controlled risk analysis. To simulate an autonomous driving scenario, we define as *high-risk cases* all those instances where the ground-truth distance is smaller than the predicted one, hence a collision is more likely to happen. We estimate that among the 1932 detected pedestrians in KITTI which match a ground-truth, 48% of them are considered as *high-risk cases*, but for 89% of them the ground-truth lies inside the predicted interval.

Challenging Cases. We qualitatively analyze the role of the predicted uncertainty in case of an outlier in Figure 3.8. In the top image, a person is partially occluded and this is reflected in a larger confidence interval. Similarly in the bottom figure, we estimate the 3D localization of a driver inside a truck. The network responds to the unusual position of the 2D keypoints with a very large confidence interval. In this case the prediction is also reasonably accurate, but in general an unusual uncertainty can be interpreted as a useful indicator to warn about critical samples.

We also show the advantage of estimating distances without relying on homography estimation or assuming a fixed ground plane, such as [39, 40]. The road in Figure 3.8 (top) is uphill as frequently happens in the real world (*e.g.*, San Francisco). MonoLoco++ does not rely on ground plane estimation, making it robust to such cases.

Ablation Studies. In Table 3.3, we analyze the effects of choosing a top-down or a bottom-up pose detector with different loss functions and with our deterministic geometric baseline. We compare our Laplace-based $L1$ loss of Eq. 3.4 with a relative L_1 loss

$$L_1(x|d) = |1 - d/x| \quad , \tag{3.7}$$

and a Gaussian loss

$$L_{\text{Gaussian}}(x|d,\sigma) = \frac{(1-d/x)^2}{2\sigma^2} + \frac{1}{2}\log(\sigma^2) \quad . \tag{3.8}$$

The Gaussian Loss is based on the negative log-likelihood of a Gaussian distribution and corresponds to an L_2 loss attenuated by a predicted σ in the location. Intuitively, L_2 type losses are more sensitive to outliers due to their quadratic component. All the losses make use of relative distances for consistency with Eq. 3.4. From Table 3.3, we observe that L_1-type losses perform slightly better than the Gaussian loss, but the main improvement is given by choosing PifPaf as pose detector.

A run time comparison is shown in Table 3.4. Our method is faster or comparable to all the other methods, achieving real-time performance.

Table 3.4 – Single-image inference time on a GTX 1080Ti for KITTI dataset [64] with OpenPifPaf [94] as pose detector. We only considered images with positive detections. Most computation comes from the pose detector (ResNet 50 / ResNet 152 backbones). For Mono3D, 3DOP and MonoPSR we report published statistics on a Titan X GPU. In the last line, we calculated epistemic uncertainty through 50 sequential forward passes. In future work, this computation can be paralleled.

Method \ Time [ms]	t^{pose}	t^{model}	t^{total}
Mono3D [39]	-	1800	1800
3DOP [40]	-	2000	2000
MonoPSR [99]	-	200	200
Our MonoLoco++ (1 forward pass)	89 / 162	10	**99 / 172**
Our MonoLoco++ (50 forward passes)	89 / 162	51	140 / 213

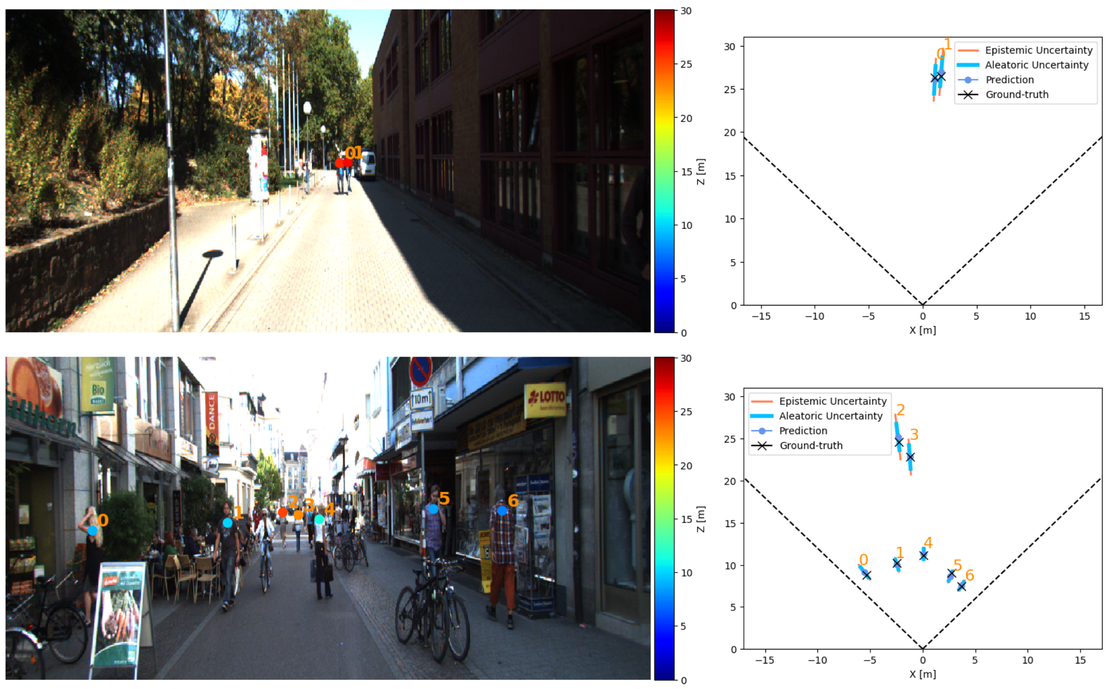


Figure 3.6 – Illustration of results from KITTI [64] dataset containing true and inferred distance information as well as confidence intervals. The direction of the line is radial as we use spherical coordinates. Only pedestrians that matches a ground-truth are shown for clarity.

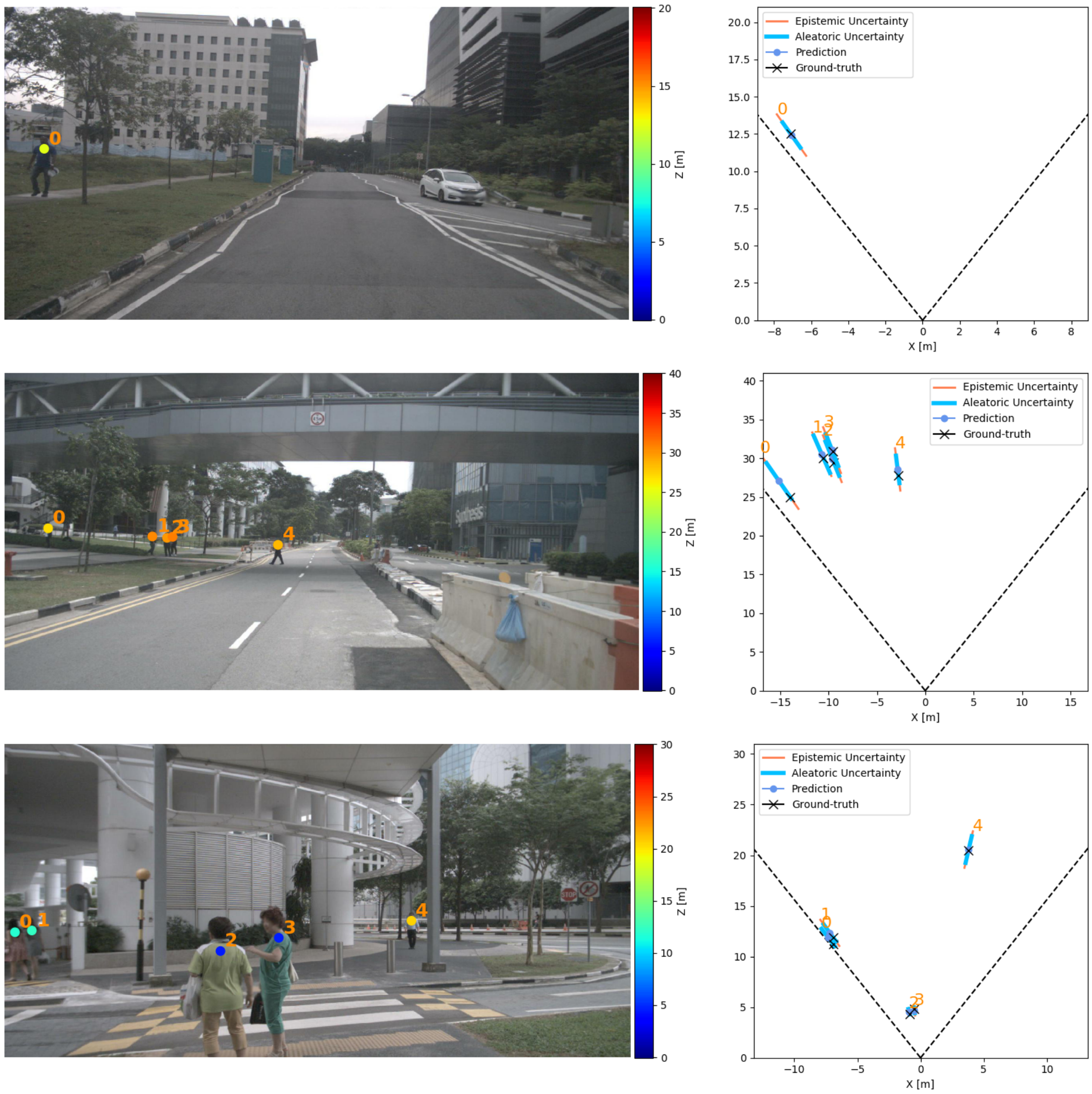


Figure 3.7 – 3D localization task. Illustration of results from nuScenes dataset [31] containing true and inferred distance information as well as confidence intervals.

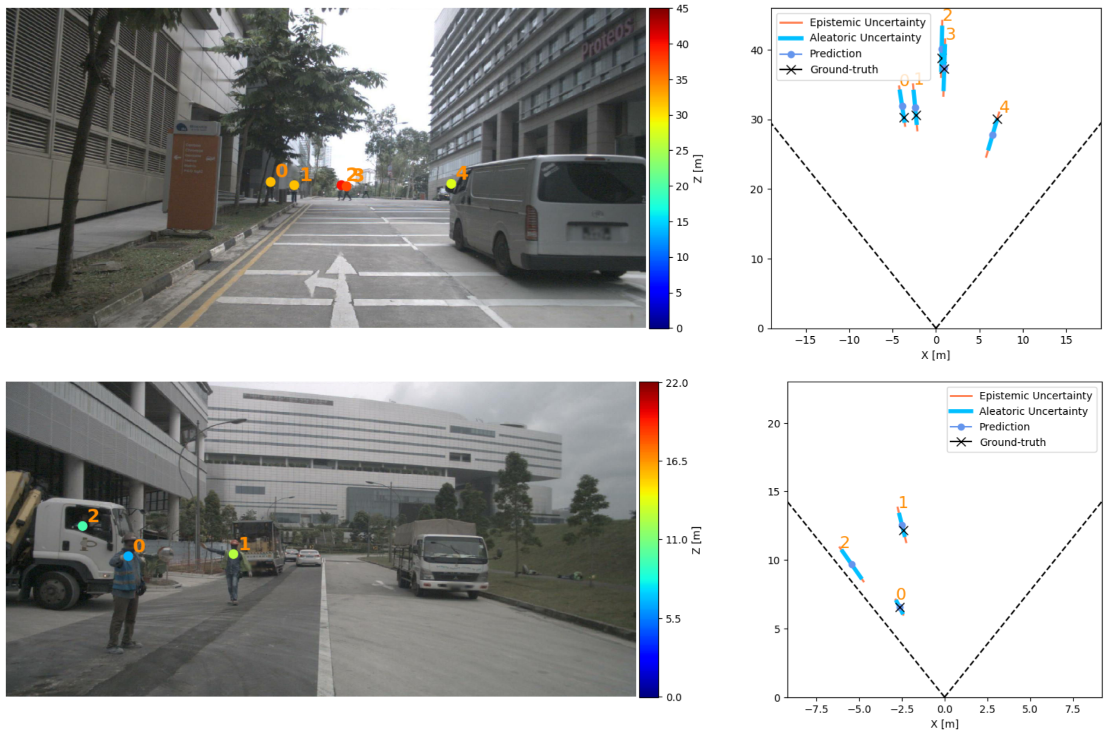


Figure 3.8 – These examples show 1) why relying on homography or assuming a flat plane can be dangerous, and 2) the importance of uncertainty estimation. In the top image, the road is uphill and the assumption of constant flat plane would not stand. MonoLoco++ accurately detects people up to 40 meters away. Instance 4 is partially occluded by a van and this is reflected in higher uncertainty. In the bottom image, we also detect a person inside a truck. No ground-truth is available for the driver but empirically the prediction looks accurate. Furthermore, the estimated uncertainty increases, a useful indicator to warn about critical samples.

3.6 Negative Results: Correlating Distance and Height Estimation

This chapter hypothesizes that the ill-posedness of the 3D localization task, while inevitable, does not prevent accurate 3D localization. We also highlight how the human height variation implicitly causes the 3D localization error through the task error of Eq 3.2. Then, our next research question becomes: "can we improve our 3D localization estimates by consistently predicting a person's height and distance?". For example, the same set of 2D keypoints may refer to a shorter person closer to the camera or to a taller person further from the camera.
In this Section, we exploit the strong correlations between a person's height, the appearance in the image plane, and the distances from the camera. We specifically refer to Zamir *et al*. [211], where they demonstrate that neural networks do not make consistent predictions without enforcing additional constraints. Hence, they develop a cross-task consistent learning procedure to improve the predictions. The proposed method is based on the concept of inference-path invariance, where "*the result of inferring an output domain from an input domain should be the same, regardless of the intermediate domains mediating the inference*" [211].

We develop a cross-task consistency (X-TC) based training procedure to consistently learn people's height and their distance from the camera. Our scenario differs from the proposed ones in [211], where the inputs and the outputs are high-dimensional, *e.g*., from raw images to surface normals or dense depth estimation. In this approach, the inputs are low-dimensional 2D keypoints and the output instance-based attribute of pedestrians: distance and height. In addition, we combine cross-task consistency with multi-task learning, where each of the networks may simultaneously predict multiple tasks, *e.g*., estimating the radial distance from the camera together with the horizontal and vertical location in the 3D world. Finally, in our scenario, the relation between height and distance is conditioned to the appearance of the person in the image plane, and we include this conditioning in our X-TC architecture, as shown in Figure 3.9.

3.6.1 Consistency Functions

To ensure consistency between distance and height, we study an analytical approach as well a a neural network approximation. In both cases, the distance in meters cannot be informative regarding the height without including additional information. Hence, we combine the distance with the height of the person in the image plane (in pixels) to obtain the real height in meters.

Analytical Function. In its simplest form, the analytical relation between the variables is the following:

$$h = C * H * d \quad , \tag{3.9}$$

where H is the 2D bounding box height in pixels, d and h the distance and the person height in meters. C is a constant that can be derived having access to the ground-truth heights and

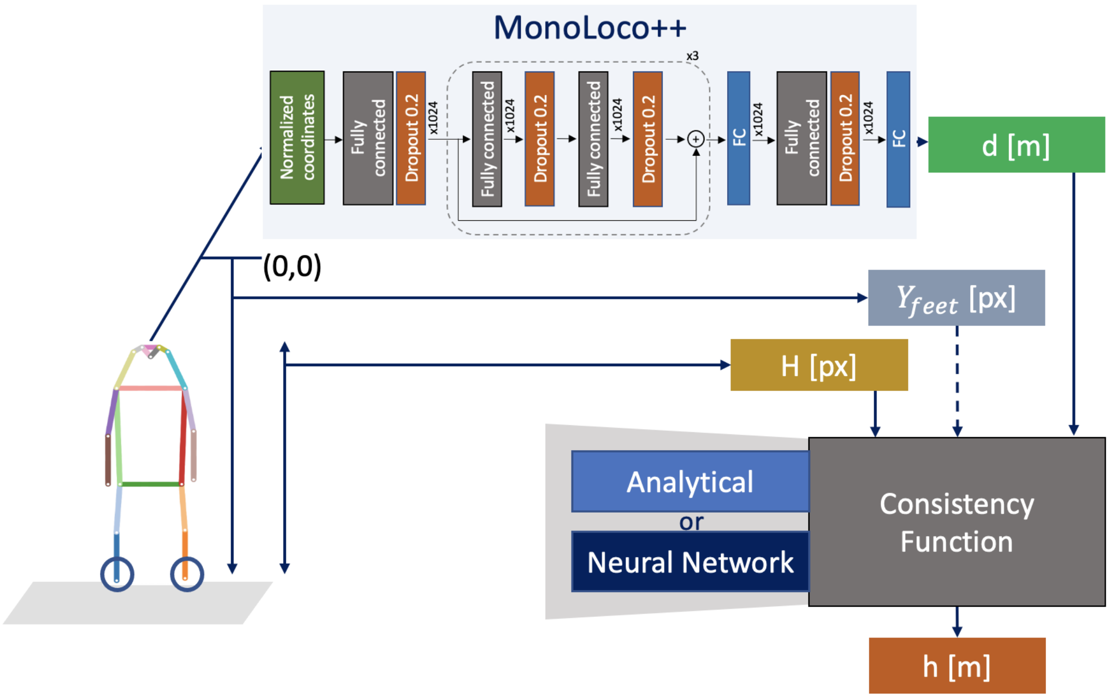


Figure 3.9 – Cross-task Consistency (X-TC) architecture: the input of our keypoint-based model is the set of 2D body joints extracted from a raw image, and the output is the distance of the person from the camera. The consistency function receives as input the estimated distance d, as well as the height of the person in the image plane (in pixels) H, and the vertical location of the feet Y_{feet}, and estimates the real height of the person h (in meters).

distances.

Neural network. Alternatively, we train a simple neural network, consisting of two blocks of fully connected layers to derive h given d and H. In addition, we include the vertical location of the feet of the person, Y_{feet}, as additional input. We use it as a proxy to inform the network about the location of the person in the ground-plane. Assuming a flat ground-plane, the network can learn to use Y_{feet} to better correlate d and h in case of noisy predictions. An overview of the consistency function is shown in Figure 3.9.

Triangle Loss. We implement the triangle loss described in [211], using distance and the height as our target domains. We use as direct losses a Laplacian-based loss function for the distance [85], and a L1 loss for the height estimation. Our triangle loss is the following:

$$L_{triangle} = L_d + L_h + |i(f(x)) - g(x)| \quad , \tag{3.10}$$

where L_d and L_h are the direct losses, x the set of input keypoints, f, g and i neural networks estimating distance, height, and consistency.

Separable and Perceptual Loss. The triangle loss, while intuitive, has the following limitations: (i) it requires simultaneous training of all the output domains, and (ii), it assumes a perfect function for mapping from the height to the distance. These assumptions are not verified in our scenario, as the height in pixels H is provided by the off-the-shelf pose detector, which we can consider as a noisy sensor. The person height h is noisy too, as we are approximating it using the height of the enclosing 3D box.

If we are only interested in predicting the distance d, we can convert the triangle loss into a simpler loss in the form of:

$$L_{separable} = L_d + |i(f(x)) - h_{gt}| \quad , \tag{3.11}$$

by substituting the neural network g with the ground-truth height. However, the link from $i(f(x))$ to h_{gt} may still be ill-posed, resulting in a non-zero loss, and when optimizing jointly the networks, the error may corrupt the optimization of the network f (the one that estimates the distance directly), leading to a degradation of performances.

We follow [211] and handle this by adopting a *perceptual loss* that depends on high-level features from a pre-trained loss network. In particular, we compare $f(x)$ against the ground-truth distance d_{gt}, through the lenses of the network i. The perceptual loss has the following formulation:

$$L_{perceptual} = L_d + |i(f(x)) - i(d_{gt})| \quad , \tag{3.12}$$

This trick avoids the residual imperfections of $i(d_{gt})$ to be back-propagated during the training phase and to corrupts the training of $f(x)$. The overall architecture trained with the perceptual loss is shown in Figure 3.9.

3.6.2 Cross-Task Consistency Evaluation

X-TC Training Procedure When training our model with the X-TC pipeline, we include the following steps:

- initialize each network with standard direct training
- select the best validation epoch for each of the network and store the trained models
- train the main network (for distance estimation) using one of the following losses:

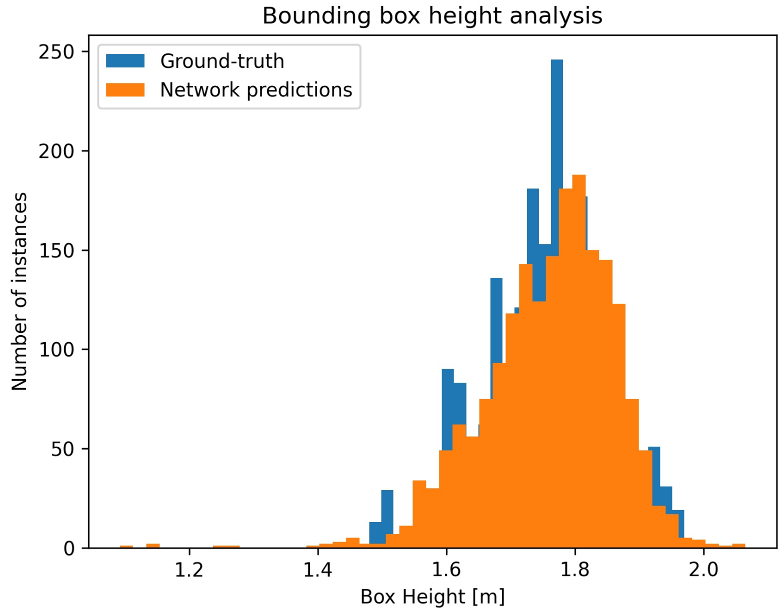


Figure 3.10 – Distribution of human heights predicted by our network trained using 2D keypoints compared with the ground-truth distribution for the KITTI validation set. The average error is 8.5 cm.

1. Triangle loss of Equation 3.10
2. Separable loss of Equation 3.11
3. Perceptual loss of Equation 3.12

and select the the best results among the following two options:

- freezing all the parameters of the (pre-trained) consistency networks and only train the network for distance estimation
- training all the parameters of all the two/three networks together

Finally, to balance the losses, we apply a scaling factor to translate a distance error into a height error. In the simplest version, we assume an average distance of 15 meters and scale the loss terms related to the height of this scaling factor.

The X-TC pipeline can be seen as a stand-alone addition to our method. Thus, we first assess our approach without X-TC consistency and add an ablation study in Section 3.6.2 to evaluate the performances of our method when correlating height and distance estimation.

Height Estimation Results. Our goal is to improve the 3D pedestrian localization performances by correlating these results with the height estimation ones. However, a neural network could simply learn the average height for each pedestrian instead of its variations. We verify that the network estimates are not constant and in Figure 3.10, we show the distribution of the ground-truth height for the KITTI validation set compared with the one predicted by the network. Our network estimates the 3D bounding box height with an average error of 8.5 cm, and the distribution of its prediction overlaps the ground-truth one. We are not claiming that the network can correctly predict a person's height, and a residual error remains as expected. However, we are still interested in studying whether we can improve the overall performances by correlating height and distance estimation and their errors.

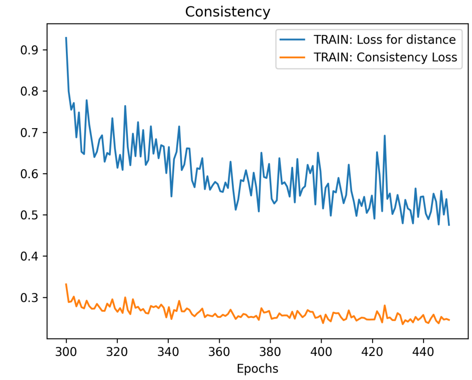
Table 3.5 – Comparing our X-TC pipeline with different loss functions. *no X-TC* is the MonoLoco++ model retrained with the same hyper-parameters of the X-TC models, and without orientation estimates, to ensure fair comparison. All approaches achieve approximately on par results. Among them, our X-TC model trained with perceptual loss shows marginally superior performances on the ALA metric.

Method	ALE (m) ↓ [Recall (%) ↑]				ALA (%) ↑		
	Easy	*Mod.*	*Hard*	*All*	$< 0.5m$	$< 1m$	$< 2m$
MonoLoco++ [24]	0.69 [90%]	**0.71** [66%]	1.37 [31%]	0.76 [70%]	37.4	53.2	63.6
No X-TC	0.70 [91%]	0.82 [68%]	1.07 [32%]	0.76 [71%]	**37.9**	54.0	**64.7**
X-TC Triangle	**0.64** [87%]	0.78 [63%]	1.17 [27%]	**0.72** [67%]	36.4	52.6	61.9
X-TC Separable	0.72 [90%]	0.76 [65%]	**1.07** [31%]	0.76 [69%]	37.0	52.4	63.5
X-TC Perceptual	0.70 [91%]	0.77 [66%]	1.11 [31%]	0.75 [70%]	**37.9**	**53.6**	**64.4**

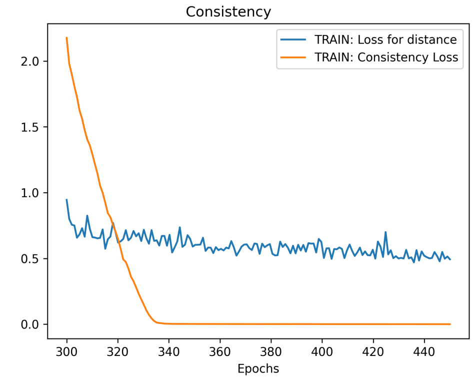
Consistency Network Results We test our consistency function that estimates a person's height starting from his/her 3D distance. We either use the analytical formulation of Equation 3.9, or a simple fully connected network.

With the analytical function, we obtain a constant $C = 0.1156$ with a relative standard deviation of 13.6 %. We calculate this using the ground-truth distance and the 2D bounding box height in pixels from the pose detector. When using the ground-truth 2D box from the KITTI dataset [64], the relative standard deviation lower to 8.0 %. It is worth noticing that the ground-truth 2D box (i) cannot be used during evaluation, (ii) it is still not precise, as it is the projection of the ground-truth 3D bounding box. We, therefore, use the constant obtained from the bounding box enclosing the 2D keypoints, and we obtain an average error of 14.9 cm. The error is not negligible considering that the standard deviation of human height is around 9 cm [194]. However, the analytical function is a simple approximation that considers a pinhole camera and neglects that we are estimating the height of the 3D bounding box enclosing the pedestrian instead of the actual height.

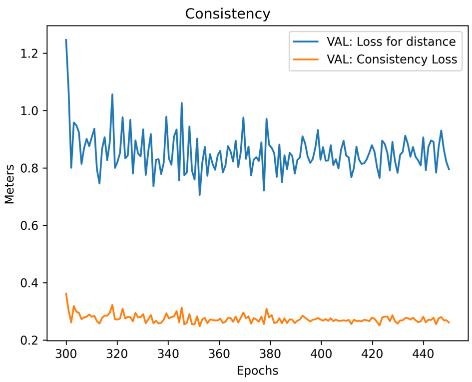
Alternatively, we train a simple network with two fully-connected blocks with a fully connected layer (FC), a Batch Normalization layer (BN) [79], and a ReLU activation function. We feed the network with the ground-truth distance and the bounding box height from the 2D keypoints, obtaining an error of 7.08 cm on the validation set of KITTI. When adding the information on the y-location of the feet of the pedestrian (as a proxy for the ground-plane), our results marginally improve to 6.28 cm. These results highlight the amount of noise between the person's height and his/her distance when conditioning to the person's appearance in the image plane.



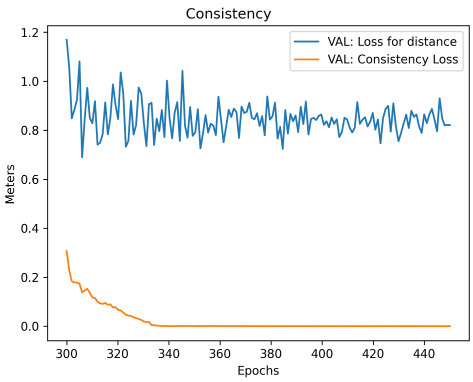
(a) Training loss. Freezed consistency network parameters.



(b) Training loss. Including consistency network parameters.



(c) Validation loss. Freezed consistency network parameters.



(d) Validation loss. Including consistency network parameters.

Figure 3.11 – Training and validation loss functions using the perceptual loss of Equation 3.12 and the architecture in Figure 3.9. The blue line represents the direct loss for distance estimation, the orange line the consistency component of the loss, including a scaling factor of 15 to translate height errors into distance errors. The validation losses concerning the direct training from the 2D keypoints to the distance do not improve and the training losses tend to overfit. Performances are not affected by the scaling factor.

3D Localization Results In Table 3.5, we compare our MonoLoco++ trained without X-TC with three versions of our X-TC pipeline, using respectively the triangle, the separable, and the perceptual losses. Our results are comparable in all the above cases. One explanation is that the amount of noise inside the consistency function described in the previous paragraph may prevent further improvements. As seen in Figure 3.10, even if the network estimated the distribution of human heights from 2D keypoints, the link between the ground-truth height and distance remains noisy. More details are shown in Figure 3.11, where we show the training and validation losses when using the perceptual loss of Equation 3.12. The loss components that account for the direct training of the distance from keypoint do not improve in the validation set, showing overfitting in the training set.

In summary, exploiting the correlation between humans' height and their distance from the camera does not result in any substantial improvement regarding the 3D localization task. However, the study provides insights about directly estimating humans' height and the amount of noise between a person's height and his/her distance from the camera.

3.7 Conclusion

We presented a new deep learning method that uses monocular cameras to perceive humans' 3D location and emphasized that the main challenge is the intrinsic ambiguity of the task. Thus, we presented a method that predicts calibrated confidence intervals in contrast to point estimates leading to state-of-the-art results in the 3D localization task.

While monocular cameras find application in many different sectors, it is fundamental to minimize uncertainty and maximize the prediction accuracy in autonomous driving applications. The next chapter will describe how to best leverage semantic keypoints with stereo cameras. We will show that estimating the 3D localization of people from stereo cameras presents a separate set of challenges and demonstrate how to obtain robust and accurate predictions by combining monocular and stereo cues.

4 Tackling the Long Tail of 3D Pedestrian Localization with Stereo Cameras

This chapter is based on the articles:

Lorenzo Bertoni, Sven Kreiss, Taylor Mordan, and Alexandre Alahi, *Monstereo: When Monocular and Stereo Meet at the Tail of 3D Human Localization*, The IEEE International Conference on Robotics and Automation (ICRA), 2021

Deng Wenlong, Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *Joint Human Pose Estimation and Stereo 3D Localization*, The IEEE International Conference on Robotics and Automation (ICRA), 2020

Additional qualitative results are displayed on YouTube[I].

4.1 Introduction

Monocular and stereo visions are cost-effective solutions for 3D pedestrian localization in the context of self-driving cars or social robots. In the previous chapter, we have studied monocular vision, describing its strengths and limitations. This chapter will focus on perceiving humans from stereo cameras, proposing a novel unified learning framework that leverages the strengths of both monocular and stereo cues to focus on the long tail of 3D pedestrian localization.

Stereo cameras have been extensively used as a cost-effective sensor to 3D locate humans in the context of autonomous vehicles or social robots [10, 99, 23, 48, 197], and all the most recent approaches strive to improve state-of-the-art results in popular metrics. Yet these solutions do not necessarily convey trust in real-world applications, and the long tail of 3D perception opens a Pandora's box of undetected challenges. While many methods perform very well "on average", can they still be trusted in the most challenging cases? The long tail of 3D object localization, *i.e.*, the share of instances where methods struggle the most, is crucial for safety but rarely evaluated in standard benchmarks [64]. This is especially relevant for pedestrians, undoubtedly an essential

[I] https://www.youtube.com/watch?v=pGssROjckHU

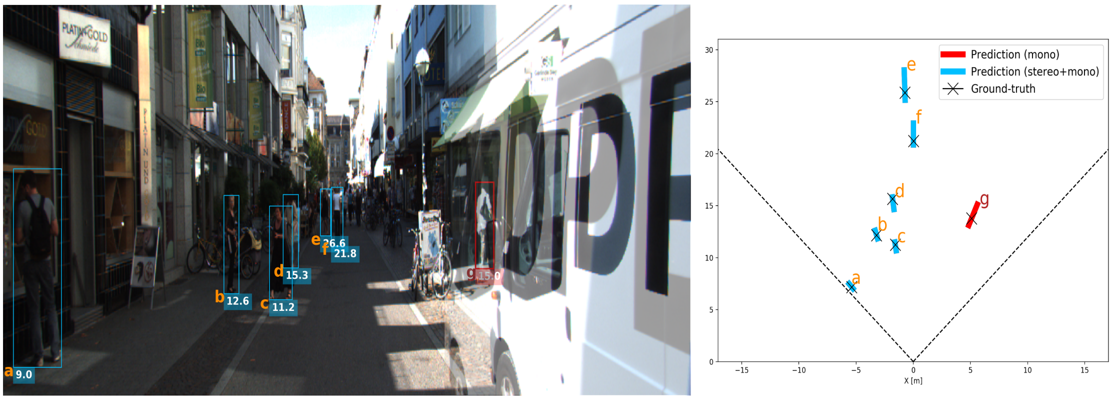


Figure 4.1 – Long tail example in the KITTI dataset [64]. The pedestrian *g* is only visible from the left camera (no stereo information available) as shown by overlapping the white van from the right image. The network classifies it as a monocular sample (red color) and outputs a larger confidence interval that reflects less accurate monocular estimates at that location. We display radial distances in meters in the frontal image and radial uncertainties in the bird-eye-view image. Only instances that match a ground-truth are shown.

category to safeguard from vehicle or robot collisions.

Multi-view [4, 8, 7, 11] and stereo-based [195, 48] methods have the potential for accurate 3D human localization, as they are free from the perspective projection ambiguity, inevitable in the monocular case [23]. Pseudo-LiDAR [197] drastically reduced the discrepancy between camera and LiDAR performances by converting a stereo-based dense depth map into 3D point clouds and directly applying LiDAR-based object detectors [149, 98]. However, computing depth from disparity poses two main challenges. Instances can be located out of the field of view or be occluded in one of the two images, and an association may not be available. Furthermore, a small disparity error (*e.g.,* a pixel shift) for far-away objects leads to unacceptable errors of several meters, as the error grows quadratically with depth [62]. We identify occluded and far instances as the largest share of the stereo-based long tail of predictions. On the contrary, monocular images are less error-prone for far instances and do not depend on accurate detections on both images. In Chapter 3, we achieved competitive performances in 3D human localization by exploiting the known prior distribution of human heights. However, this approach fails in the presence of children or very tall people, connecting the long tail of monocular 3D localization with the distribution of human heights.

In this chapter, we leverage the best of both worlds, *i.e.*, stereo and monocular methods, in a unified learning framework tailored for pedestrian 3D localization. Our method, referred to as *MonStereo* [25], jointly associates detections in left-right images and implicitly learns to leverage monocular and/or stereo cues. Moreover, it also learns to communicate uncertainty driven by the cues (again without direct supervision at training time). Our approach uses an off-the-shelf pose detector [94] on left-right images to obtain 2D keypoints, a low-dimensional representation of humans. A simple feed-forward network estimates whether each input pair is formed by the same person from left-right images and, concurrently, estimates the 3D location of pedestrians with their corresponding uncertainty (accounting for stereo disparity and/or monocular cues).

The popular KITTI dataset [64] has a limited variation of instances, oversimplifying the monocular task. We address the long tail of height distribution by injecting prior knowledge from the real world. Leveraging the simplicity of manipulation of 2D keypoints, we create instances of people from a broader spectrum of heights. This conveys information about the real challenge of the task in the data domain, thus, increases the network performance and calibrates the estimated confidence intervals without the need for hand-crafted architectures.

In summary, we propose a unified learning framework that jointly matches detections in left-right pairs of images and estimates the 3D localization of each pedestrian. We focus on the limitations of monocular and stereo visions, referred to as the long tail challenge, by jointly exploiting stereo and monocular cues with a measure of uncertainty. We also design a data augmentation procedure to tackle the long tail of the human height distribution. Our network achieves state-of-the-art results on 3D localization metrics and provides reliable confidence intervals even for challenging cases. The code[II] and a video[III] with qualitative results are available online.

4.2 Related Work

Stereo 3D Detection. Stereo-based 3D detectors can be grouped into *instance-level* and *pixel-level* depth estimators. The *instance-level* approach consists of detecting instances in the image plane and comparing features of proposals in left and right frames to correctly associate objects and estimate their locations [40, 107, 106, 152, 48, 147]. Among them, PSF [48] was designed for human localization and, similarly to our method, leverages 2D keypoints to solve the association task. However, their 3D output is simply the median depth calculated from a set of disparities. The *pixel-level* approach consists in estimating a dense disparity map for every pixel and transforming the dense map into a 3D point cloud [197, 150]. The pseudo-LiDAR point cloud can then be used to detect vehicles and pedestrians by applying LiDAR-based algorithms [149, 98]. The underlying task of all previous methods is to compute disparity from pixels, either locally to associate and align pairs of left-right instances, or globally to find dense correspondences between pixels. Qin *et al.* [152] have recently proposed to extend a monocular baseline to predict 3D locations of *car* instances with a triangulation network. Our work goes beyond the concept of "depth from disparity" and is not limited by the discrete nature of pixels, but can exploit together monocular and stereo cues to directly estimate a continuous depth.

Vision-based 3D Localization Ambiguity. Estimating the 3D location of objects from a single RGB image is a fundamentally ill-posed problem due to the ambiguous projections to the 2D image. This is particularly true for humans due to their variation of height and non-rigid body

[II] https://github.com/vita-epfl/monoloco
[III] https://www.youtube.com/watch?v=pGssROjckHU

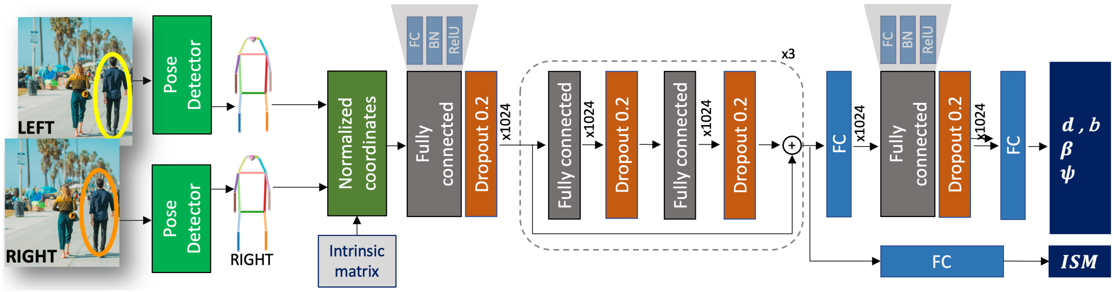


Figure 4.2 – Network architecture. The input is a set of 2D keypoints extracted from a raw image. The outputs are the radial distance d with its confidence interval b, the azimuthal angle β, the polar angle ψ, and the Instance-based stereo matching (ISM). Every fully connected layer is followed by a Batch Normalization layer (BN) [79] and a ReLU activation function.

structure. In Chapter 3, we quantified this ambiguity as a function of the distance from the camera, assuming that the distribution of human stature follows a Gaussian distribution for male and female populations [59]. The expected localization error $\hat{e}_{mono}$ due to height variations of people can be obtained by $\hat{e}_{mono} = C * d_{gt}$, where the constant C is modelled from the distribution of human heights and d_{gt} is the ground-truth distance. On the other side, even if stereo methods do not suffer from the intrinsic ambiguity of perspective projection, the error grows quadratically with depth, making disparity estimation very sensitive to pixel resolution. The depth error e_z can be expressed as a function of the disparity error e_d [62] as $e_z \approx \frac{z^2}{bf} e_d$, where z is the depth, b the camera baseline and f the focal length. With the goal of comparing monocular and stereo limitations, we analyze what we call the *pixel error*: the depth error due to a disparity error of one pixel. Its value depends on the characteristics of the camera and we use the camera parameters of the KITTI dataset [64], a popular dataset for 3D object detection with stereo imaging at a resolution of 1240×380 pixels. The results, shown in Figure 4.5, highlight that the stereo depth error can become more challenging than the monocular one for humans at just over 20 meters. For example, a disparity error of 1 pixel at 40 meters corresponds to 4.5 meters of depth error. These conclusions depend on the precision of the disparity estimation and the image resolution, but highlight the importance of monocular estimation for 3D perception.

4.3 Method

Our approach aims to detect, associate and estimate the 3D positions of pedestrians in a pair of stereo images. We identified two main challenges for a stereo network: (i) when a person is not identified in both images, there is no disparity information, and (ii) disparity estimation for faraway objects leads to poor predictions. We propose a simple yet effective way to tackle both issues.

Architecture. Our method consists of two steps. First, we reduce the input dimensionality by predicting 2D keypoints for each person in left-right images. Keypoints are a low-dimensional representation which is invariant to many nuisances, is suitable in the low-data regime and is prone to easy manipulations. Second, we analyze pairs of keypoints from left-right images in an "all-vs-all" setting to predict 3D location, and confidence interval of every person in the scene. Our simple architecture is shown in Figure 4.2 and consists of few fully-connected layers with batch-normalization, residual connections [73], and dropout [183].

Input/output. Similarly to our monocular architecture [24], we use the pose detector described in Chapter 2 to obtain a set of keypoints $[\vec{x}, \vec{y}]_i^T$ for every person i in the left and right images. Each keypoint is projected into normalized image coordinates $[\vec{x}^*, \vec{y}^*, \vec{1}]_i^T = I^{(i)}$ to prevent overfitting to a specific camera. To construct the network inputs, we associate in an "all-vs-all" way the keypoints $I^{(l)}$ from each person l in the left image with the one $I^{(r)}$ from each person r in the right image, to form the associated pair $I^{(l,r)}$:

$$I^{(l,r)} = I^{(l)} \parallel (I^{(l)} - I^{(r)}) \;\; \forall \, l \in N_L, r \in N_R \quad , \tag{4.1}$$

where $\parallel$ is a concatenation operation, and N_L, N_R denote the sets of detected instances in the left-right image pair. If the sets of keypoints $I^{(l)}$ and $I^{(r)}$ belong to the same person, the input is a *true pair* and we call it $I^{(l,r)}_{S-M}$, with subscript denoting stereo and monocular cues available. Otherwise the input is a *false pair*, which we denote with $I^{(l,r)}_M$. We treat this problem as a binary classification task and use binary cross-entropy loss to train our network. We refer to this association task as *Instance-based stereo matching (ISM)* and to its loss as *ISM loss*. As in Chapter 3, to disentangle the depth ambiguity from the other localization components (x, y), we use a spherical coordinate system (d, β, ψ), namely radial distance d, azimuthal angle β, and polar angle ψ.

Uncertainty. We model the aleatoric uncertainty for the depth estimation task using the relative Laplace loss function described in Section 3.4. At inference time, the model predicts a radial distance d and a spread b which indicates its confidence about the predicted distance. The use of spherical coordinates allows to convey all the 3D localization uncertainty into the radial component d.

Inference. The network performs 3D localization as well as ISM by predicting whether each pair of keypoints belongs to the same person or to different ones. The ISM component is also used to filter multiple results for the same person. At inference time, the network predicts N_R outputs for each person in the left image (one for each associated pair) and selects the one with the highest predicted stereo matching. In fact, a *true pair* $I^{(l,r)}_{S-M}$ always contains more information about the left instance than $I^{(l,r)}_M$. For a single image pair, the number of pairwise combinations grows quadratically as $N_L * N_R$ but, as the inputs are low-dimensional, the computation is parallelizable by including all the pairs in the same batch.

Knowledge Injection. Monocular estimates are essential to address the long tail of stereo-based 3D localization, but they present their own issues. A typical dataset for 3D object detection such as KITTI [64] is not representative of the real world as it only contains few scenes recorded from a single city. For instance in the case of a child, any monocular estimate of depth will either fail or rely solely on the ground plane estimation [50]. These settings make the network over-confident toward monocular estimates, as: (i) the predicted confidence intervals do not reflect the real distribution of human heights and the model can drastically fail in case of children or tall people; (ii) the training phase becomes ineffective as the network relies on monocular estimates even when a stereo association is available. To tackle both issues, we inject knowledge in the training data by augmenting it with relevant examples from the long tail of the human height distribution. We augment the KITTI dataset with synthetic 2D keypoints of people of heights ranging from 1.2 meters to 2 meters. We rely on the mild assumption that the aspect ratio between children and adults is unchanged and for each set of keypoints $I^{(l,r)}$, we sample a height h from the uniform distribution $\mathscr{U}(1.2, 2)$ and we derive a new ground-truth distance from the triangle similarity relation of human heights and distances. Then, we create a new input $I^{*(l,r)}$ updating the disparity and the ground-truth distance. We repeat this procedure for every pair $I^{(l,r)}$ with double-sided advantages. The network benefits from augmenting the number of *true pairs* $I_{S-M}^{(l,r)}$ as it learns that disparity estimates correspond to correct depths whereas the monocular assumption of average height breaks down. It also benefits from augmenting false pairs $I_{M}^{(l,r)}$, by becoming receptive to more realistic human height variations, including children or very tall people. This knowledge is reflected in more calibrated confidence intervals of distance especially in the tail of human heights.

4.4 Alternative Methods

Our MonStereo receives as input sets of keypoints obtained from an off-the-shelf pose detector and solves the stereo matching and 3D localization tasks together. As alternative approaches, we propose to (i) jointly solve pose estimation and stereo matching with a single feed-forward regression network, and to (ii) separately solve the pose estimation, stereo-matching, and the 3D localization tasks.

4.4.1 End-to-end Approach

To address challenges related to keypoints' stability and limited resolution, we develop an end-to-end method, referred to as Part Spatial Fields (PSF) [48], which combines composite fields [95] and correlation layers [78]. The method reasons in 3D, creating location proposals in the form of 3D human poses. It includes a shared ResNet [73] base network and OpenPifPaf [95] head networks to predict the 2D poses and a new third head network for predicting association between stereo keypoints.

Correlation Calculation. Our goal is to detect and match across pair of stereo images multiple people at the same time. We compute correlation values for all positions in a feature map and make our model operate on the whole feature maps for matching regression. Calculating all possible circular shifts would lead to a vast output dimensionality, but stereo images do not require a significant disparity. Hence, we restrict the correlation calculation to small translations. Our correlation module is inspired by FlowNet [78], where a correlation layer is aimed to help a convolutional network in matching feature points between stereo images. The correlation layer operates pixel-level feature comparison of two feature maps x_l, x_r:

$$x_{corr}^{l,r}(i, j, p, q) = \langle x_l(i, j), x_r(i+p, j+q) \rangle, \tag{4.2}$$

where $-K \leq p \leq K$ and $K \leq q \leq K$ are offsets to compare features in the square neighborhood around the locations i, j in the feature map, defined by the maximum displacement K. The correlation layer output becomes $x_{corr} \in R^{H_l \times W_l \times (2K+1) \times (2K+1)}$. In other words, Equation 4.2 can be seen as a correlation between two feature maps within a local square window defined by K. We compute this local correlation for left regression and right regression.

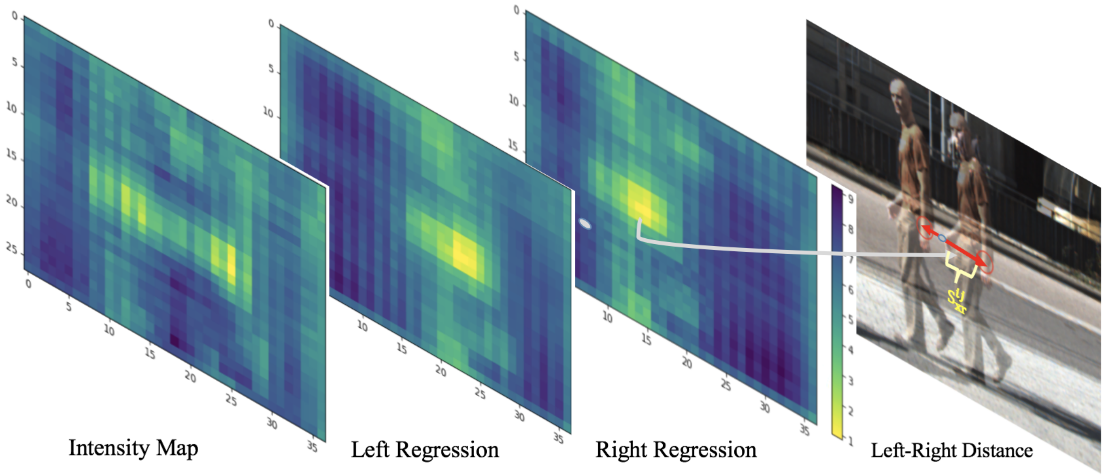


Figure 4.3 – The Part Spatial Fields (PSF) output maps for the “left wrist”. The intensity map gives the route of the joint movement. The left and right regression maps measure the down-scaled distance from each pixel location to left and right stereo keypoints. From the regression map, we generate dense association vectors to associate stereo keypoints.

Part Spatial Fields. Our Part Spatial Fields (PSF) module outputs one intensity map s_c to model the confidence of association and two regression maps $\langle s_{xl(r)}, s_{yl(r)} \rangle$ to convert the similarity into a pixel-level distance. As shown in Figure 4.3, at every output location, two vectors point to the left and right stereo keypoints locations, respectively.

ISM algorithms need to consider the diversity of scales that a human pose can have in an image. While a localization error for the joint of a close proximity person can be minor, that same absolute error might be a major mistake for faraway smaller persons. At the same time, measuring the uncertainty of the spatial precision of an association could be helpful when computing a score for each connection. As described in Section 3.4, we use a Laplace loss 3.4 to train the regressive model, and, PSF can be represented as:

$$\overrightarrow{s} = \{s_c^{i,j}, s_{xl}^{i,j}, s_{yl}^{i,j}, s_{bl}^{i,j}, s_{xr}^{i,j}, s_{yr}^{i,j}, s_{br}^{i,j}\}, \tag{4.3}$$

where (s_{xl}, s_{yl}) and (s_{xr}, s_{yr}) are absolute locations (sum of pixel location and regression distance) of an association vector's two endpoints. (s_c, s_{bl}, s_{br}) represent the confidences for the association and left and right spatial precision, respectively. The output is then decoded to associate stereo poses with high location similarity.

Finally, given the depth of each joint, we calculate the distance of the person to the camera. Since some joint locations are not accurate due to occlusion or detection error, we include a z-score thresholding procedure to remove outliers and consider the median depth of the remaining keypoints as our final output.

4.4.2 ISM Baselines.

Our MonStereo combines the ISM task with the 3D localization one. An alternative approach is to simply focus on the stereo matching task and calculate the 3D localization of each pedestrian from the given disparity. We propose three ways to associate people in left-right pairs of images.

1. *B-Median.* We use our network only for the ISM task and, if a match between two people is found, we calculate the depth from the median value of the set of disparities.
2. *B-Pose*. An intuitive way to associate people is to use pose similarity based on the detected 2D keypoints, *i.e.*, calculating how similar two poses are in left and right images. In particular, we look for the best similarity score for each person in the left image (reference) with respect to all the people in the right image (targets). We zero-center the reference and the target poses using the location of their center and we calculate the L2 norm between our reference vector and all the target vectors and save the scores. We repeat this procedure for all the poses in the left images and associate left-right pairs of poses in a greedy way.
3. *B-ReID*. We develop an image-based baseline which associates the same person in left-right pairs of images by looking at the appearance of the person and the scene around him. We use a state-of-the-art Re-Identification model [1] trained on Market-1501 [217], a very large person re-identification dataset, to make the association from cropped images.

All the three methods provide the best similarity score for each person in the left image with respect to all the people in the right image. Any association can be then either accepted or rejected. The decision is made by filters based on the coefficient of variation between keypoints and vertical disparity of the pair. If the association is accepted, the depth is calculated from disparity as described in Section 4.7.3. Otherwise, we estimate depth through a monocular estimate using MonoLoco [23].

4.5 Critical Review of 3D Metrics for Pedestrians

The majority of previous works for 3D object detections do not report results on the pedestrian category [163, 106], even when using LiDAR point clouds [98]. We argue that the most common metrics for 3D object detection, namely the bird's eye view (BEV) and 3D average precision metrics [64], are not representative of effective performances in case of pedestrians. For the BEV metric, a prediction is considered as correct if the intersection over union (IoU) between the predicted box and the ground-truth one is greater than 0.5. This metric has proven to be effective in the pixel space in popular 2D object detection benchmarks such as PASCAL VOC [54] and MS COCO [111]. On the contrary, in the 3D space a pedestrian 3D bounding box has average width and length of 60 cm and 75 cm. Considering perfect orientation and dimensions, a distance error of 18 cm already leads to an intersection over union lower than 0.5. This requirement is unnecessarily strict and shifts the attention of the community from the challenging instances to the easy ones, where obtaining results with a precision of few centimeters may still be possible. In case of stereo-based algorithms, where the disparity is large for close instances and small for further ones, this behaviour is further enforced. Furthermore, the KITTI official metric assigns to each instance a difficulty regime based on bounding box height, level of occlusion and truncation: *easy*, *moderate* and *hard*. Each category includes instances from the simpler categories, and, due to the predominant number of easy instances (1240 "easy" pedestrians and 300 "hard" ones), the metric can underestimate the impact of challenging instances. To address the current limitations, we propose to consider a safety-critical area around a pedestrian, recognizing a prediction as correct if the localization error between the predicted and ground-truth box is less than a threshold error. Differently from the metric proposed by Xiang *et al.* [203], we define an adaptive threshold based on distance. In our evaluation, we use a relative error e_z of 5%, considering 1 meter as reasonable safety-critical range for people 20 meters away, but thresholds are application dependent. We refer to the metric as Relative Average Localization Precision (RALP) and we split the evaluation into "Easy", "Moderate", "Hard", with no overlap between categories, and "All".

4.6 Experiments

4.6.1 Experimental Setup

We compare our method with the alternative approaches presented in Section 4.4. We also evaluate additional state-of-the-art monocular and stereo baselines that focus on the pedestrian category. The monocular baselines are described in Section 3.5, while the stereo ones are:

Table 4.1 – Comparing our proposed method against baselines on KITTI dataset [64]. We use OpenPifPaf [95, 94] as off-the-shelf network to extract 2D poses. On the RALP metric, our MonStereo achieves state-of-the-art results. On the ALE metric, the confidence threshold of methods has been set to 0.5 and we show the recall between brackets to insure fair comparison. Italics entries are not directly comparable as they achieve a lower recall even when no threshold is set. Our method performs better on *hard* instances while maintaining 2-5 times higher recall. The improvement of jointly solving the ISM and the 3D localization tasks is shown by the three baselines (B-).

Method	ALE (m) ↓ [Recall (%) ↑]				RALP-5% (%) ↑			
	Easy	*Mod.*	*Hard*	*All*	*Easy*	*Mod.*	*Hard*	*All*
Monocular								
Mono3D [39]	2.26 [89%]	3.00 [65%]	3.98 [34%]	2.62 [69%]	9.21	1.26	0.21	7.22
MonoPSR [99]	0.89 [99%]	2.00 [93%]	2.40 [34%]	1.51 [83%]	48.87	12.54	0.47	35.35
MonoLoco [23]	0.83 [91%]	1.12 [72%]	1.15 [27%]	0.93 [70%]	49.01	19.44	1.89	38.76
Stereo								
E2E-PL [150]	*0.12 [68%]*	*0.17 [23%]*	*0.60 [13%]*	*0.15 [43%]*	*49.32*	*4.43*	*0.44*	*31.31*
OC [147]	*0.10 [66%]*	*0.14 [31%]*	*0.75 [6%]*	*0.13 [42%]*	*65.58*	*26.38*	*1.46*	*41.30*
3DOP [40]	0.67 [88%]	1.19 [64%]	1.93 [37%]	0.93 [69%]	57.88	22.70	3.85	45.92
PL [197]	**0.16** [88%]	0.72 [59%]	1.59 [33%]	0.46 [67%]	**88.94**	42.91	**10.41**	66.33
Our PSF	0.55 [88%]	0.65 [58%]	0.80 [25%]	0.56 [65%]	57.27	19.94	4.82	46.15
Our B-ReID	0.73 [91%]	0.78 [72%]	1.02 [28%]	0.77 [70%]	73.81	39.44	4.48	58.23
Our B-Pose	0.65 [91%]	0.77 [71%]	1.18 [27%]	0.72 [70%]	73.92	39.10	4.82	58.25
Our B-Median	0.57 [92%]	0.69 [72%]	0.78 [31%]	0.61 [72%]	80.19	50.38	8.17	64.00
Our MonStereo	0.29 [92%]	**0.41** [70%]	**0.50** [31%]	**0.34** [71%]	85.54	**54.27**	8.92	**67.60**

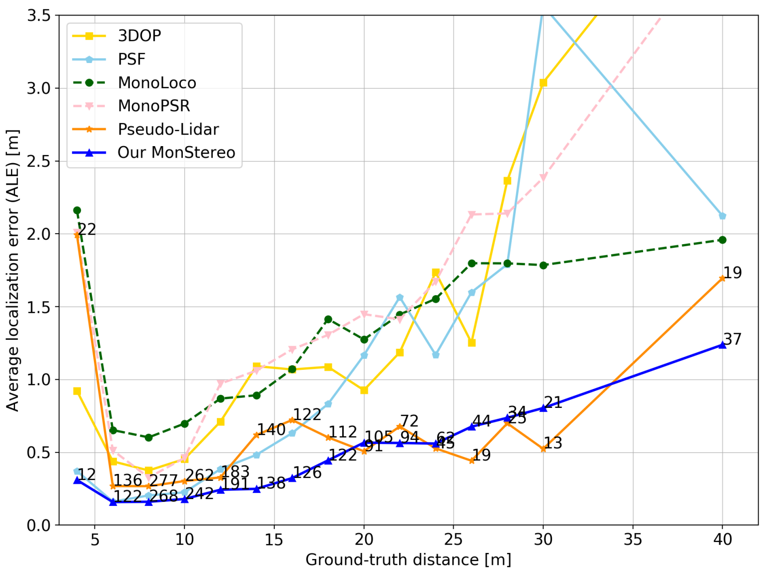


Figure 4.4 – ALE as a function of distance. MonStereo achieves robust performance while even detecting more instances (numbers included) in the farthest clusters.

- *Pseudo-Lidar* (PL) [197] converts a stereo-based dense depth map into 3D point clouds

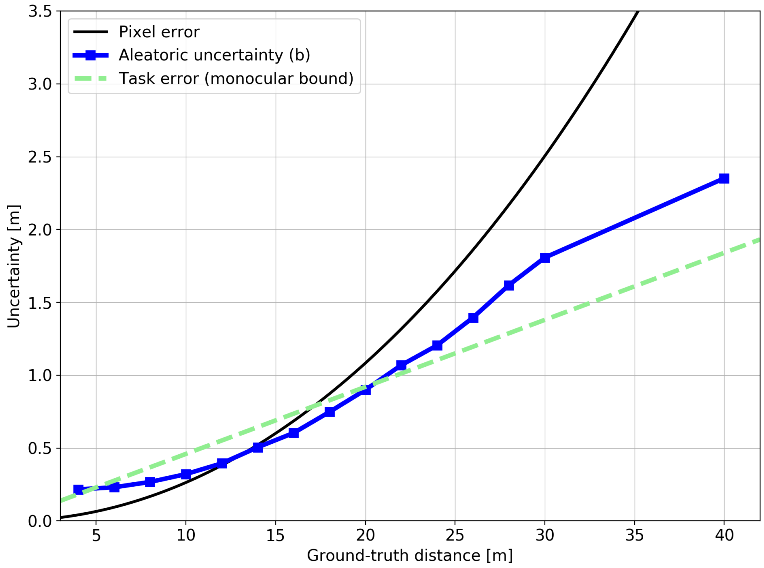


Figure 4.5 – For close instances the spread b has a quadratical trend as MonStereo exploits stereo cues, and a linear trend at further distances thanks to monocular cues.

and applies a LiDAR-based 3D object detector;

- *End-to-End Pseudo-Lidar* (E2E-PL) [150] updated framework that allows the entire Pseudo-Lidar pipeline to be trained end-to-end;
- *Object Centric (OC)* [147]: performs stereo matching on only object image crops and mask the ground truth background disparities during training to only penalize errors for object pixels;
- *3DOP* [40] is a pioneering stereo approach for pedestrians, cars and cyclists and their 3D detections are publicly available.

To evaluate pedestrian 3D object detection we use the R-ALP metric with 5% threshold. We also evaluate the Average Localization Error (ALE) [23], that differently from average precision metrics, penalizes large errors and is suited for the long tail of 3D localization.

4.6.2 Implementation Details

We train and evaluate our model on the KITTI Dataset [64] using the train/val split of Chen *et al.* [39]. To detect 2D keypoints, we use the off-the-shelf pose detector OpenPifPaf [95, 94] and we upscale the images by a factor of two to match the minimum dimension of 32 pixels for COCO instances. We train our network for 400 epochs using Adam optimizer [89], a learning rate of 10^{-3}, mini-batches of 512 and gradient clipping. We use a Laplace loss [23] for the radial distance, binary cross-entropy loss for stereo matching, and L1 loss for all other components.

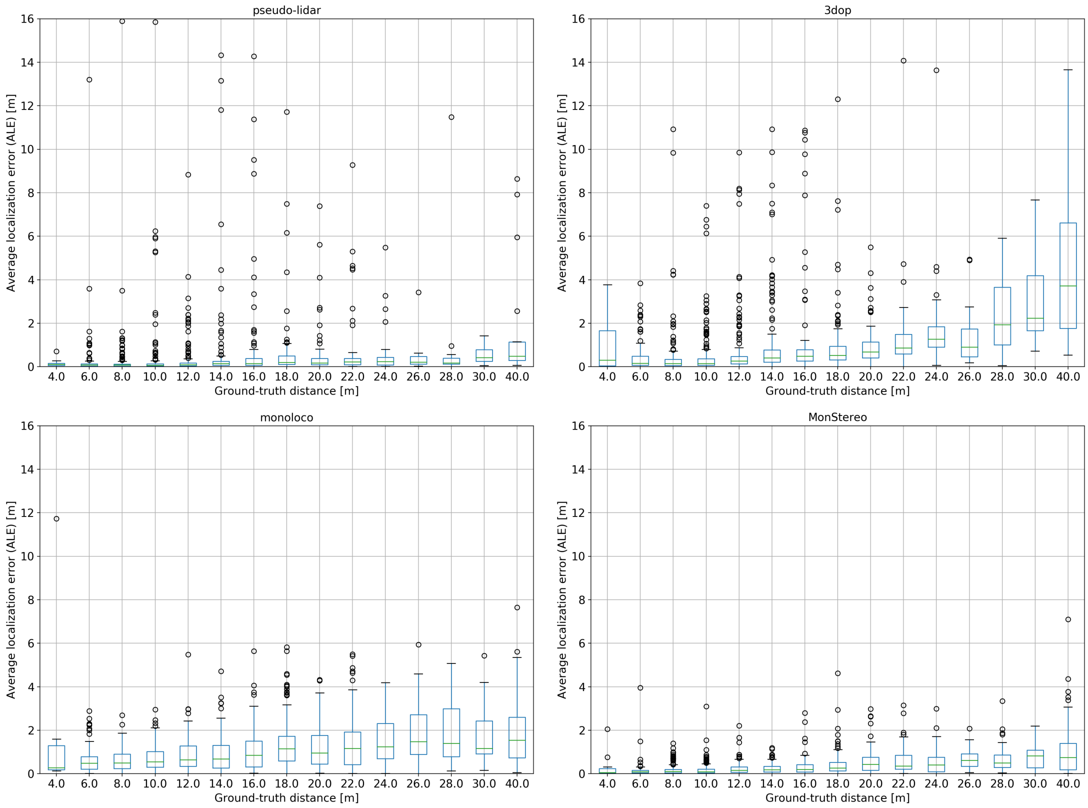


Figure 4.6 – Box plots of Average Localization Error (ALE). Circles identify outliers. Our MonStereo achieves very robust performance in the long tail with a maximum error of 7 meters for far instances and less than 5 meters in all the other cases. Every other stereo method has a few catastrophic estimates even for very close people. MonStereo's monocular component stabilizes the performances as shown by the performances of the monocular MonoLoco [23], which is on average not as accurate as a stereo method but more robust.

The losses are equally weighted. The KITTI dataset [64] does not provide pairwise matching information, thus, we extend the ground-truth by associating each person in the left image with the corresponding one in the right image (more details in Section 4.7.3). We also perform horizontal flipping and switch left and right instances.

4.6.3 Results

Table 4.1 summarizes our 3D localization results with the ALE and RALP metrics. Our method outperforms every other stereo method in the ALE metric for *Moderate*, *Hard* and *All* instances. Solving jointly the ISM task and the 3D localization one is a crucial ingredient, as shown by the three baselines where the association and localization tasks are sequential. We make in-depth comparisons with the ALE metric as a function of the ground-truth distance in Figure 4.4. On the ISM task, we obtain an accuracy of 98.2%.

Outliers. To go beyond "average-based metrics", we analyze the entire distribution of predictions through the box plots in Figure 4.6. Our MonStereo is drastically more reliable for the long tail of predictions, especially when compared to other stereo methods. MonStereo's maximum error is lower than 5 meters, while Pseudo-LiDAR [197] and 3DOP [40] have maximum errors of 24 and 17 meters respectively.

Table 4.2 – Impact of the ISM loss with mean and standard deviation (σ) of localization error. **S** simulates a standard stereo method by training a model solely with *true pairs* $I_{S-M}^{(l,r)}$; the network could learn monocular cues but is not guided by the ISM loss. **S-x** is as S, but without providing y-coordinates of input keypoints to remove information on human heights. **S+M** is trained with the same set of pairs $I_{S-M}^{(l,r)}$ and $I_{M}^{(l,r)}$ of MonStereo without the ISM Loss. The long tail of far instances is the most impacted by the ISM loss.

ALE [σ] (m)	$d < 10$	$10 < d < 20$	$20 < d < 30$	$30 < d < 50$
S	0.24 [0.6]	0.47 [0.9]	1.38 [1.4]	3.95 [2.4]
S-x	0.52 [1.5]	0.61 [1.4]	1.72 [1.5]	5.50 [3.0]
S+M	0.25 [**0.4**]	0.50 [**0.7**]	1.08 [1.2]	2.24 [1.9]
MonStereo	**0.20** [**0.4**]	**0.38** [**0.7**]	**0.73** [**1.0**]	**1.63** [**1.8**]

Long Tail-aware Confidence Intervals. The spread b is the result of a probabilistic interpretation of the model, and it is calibrated during training according to the Laplace distribution. Training data includes the long tail of the height distribution, and the number of *true* and *false* pairs is balanced by design. During validation, stereo cues are available for the majority of instances, and the confidence intervals become more conservative, with 86.0% of validation instances lying inside them. Yet the length of each side is only 3.9% of the predicted distance, making the confidence intervals small enough for practical purposes.

Monocular and Stereo Limitations. We compare in Figure 4.5 the predicted aleatoric uncertainty b with the 3D localization ambiguity for monocular and stereo modalities through the monocular *task error* and the stereo *pixel error*, respectively. The stereo ambiguity is very small at close distances but grows quadratically, while the monocular one grows linearly. Our MonStereo intrinsically learns to predict 3D locations of instances combining stereo and monocular cues based on the distance. This is reflected in the estimated confidence intervals. For close instances the trend is quadratic as stereo cues are more accurate. In contrast, at further distances the trend is linear. For very far pedestrians the predicted b is larger than the *task error* as, in addition to the task-based uncertainty, the aleatoric uncertainty b also includes input noise [85], in our case 2D keypoints noise.

Run Time. We conducted our experiments using a machine with a single NVIDIA GeForce GTX 1080 Ti and Intel(R) Core(TM) i7-8700 CPU @ 3.20GH for both 2D pose detector and MonStereo. Our run time relies heavily on the 2D pose detector [95] ($\sim$ 150 ms) with negligible

Table 4.3 – Impact of knowledge injection (KI). We trained a monocular baseline M and a stereo one without KI. Recall measures the fraction of instances inside the intervals, and *I. Size* is the ratio between the spread *b* and the ground-truth distance. KI improves performances, especially for *Hard* instances. The intervals do not grow, as the spread *b* is reduced by better exploiting stereo cues.

	ALE $\downarrow$ [m]			Recall $\uparrow$ [%]			I. Size $\downarrow$ [%]		
	Easy	*Mod.*	*Hard*	*Easy*	*Mod.*	*Hard*	*Easy*	*Mod.*	*Hard*
M	0.77	0.82	1.35	58.0	58.9	32.5	4.6	5.0	4.9
W/o KI	0.51	0.68	0.87	76.2	72.7	42.0	4.1	4.5	4.4
With KI	**0.29**	**0.41**	**0.50**	**91.2**	**81.9**	**65.4**	**3.8**	**4.1**	**4.1**

computation from 2D to 3D, making our pipeline suitable for real-time applications. The runtime is on average 1.5 ms, and can grow up to to 16 milliseconds in the most crowded case of 28/29 people in each of the two images.

4.6.4 Ablation Studies

Learning an ensemble of monocular and stereo cues is a delicate balance. The ISM loss prevents our method from overfitting to stereo disparity, and the KI prevents it from overfitting to monocular cues.

ISM Loss. This loss encourages the use of monocular cues when stereo ones are not available or monocular cues are more convenient *(e.g.*, faraway people where pixel disparity is not accurate enough). Without explicit guidance, the network over-relies on stereo cues, as shown in Table 4.2.

Knowledge Injection (KI). Without KI, the network over-relies on monocular cues. We illustrate it by training a monocular baseline, and a stereo baseline without KI. We analyze ALE metric, recall (percentage of instances inside the intervals) and relative size of the intervals in Table 4.3. KI improves results and calibrates the confidence intervals, including the long tail of the height distribution. Recall increases, yet the interval size decreases, as KI helps to exploit stereo cues and reduce the spread *b*.

4.7 Additional Results and Discussions

In this section, we show qualitative examples of challenging and critical cases from a safety point of view, and we shed light on the key challenges of our approach in these scenarios. Finally, we analyze the quality of 2D keypoints for disparity estimation using two off-the-shelf pose detectors and show that our method is agnostic to the choice of the detector.

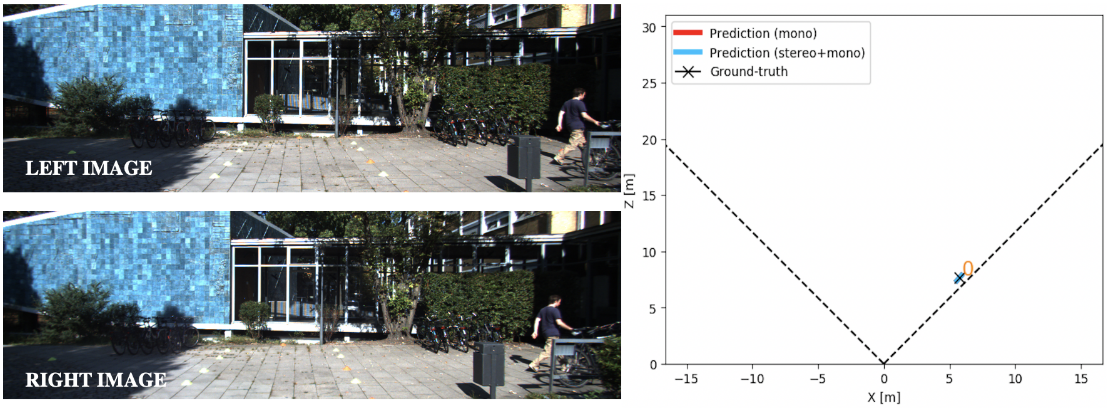


Figure 4.7 – A very close pedestrian who belongs to the *moderate* category (according to KITTI guidelines) due to the occlusion. Our MonStereo estimates accurate localization with an error of 2 cm despite the occlusion.

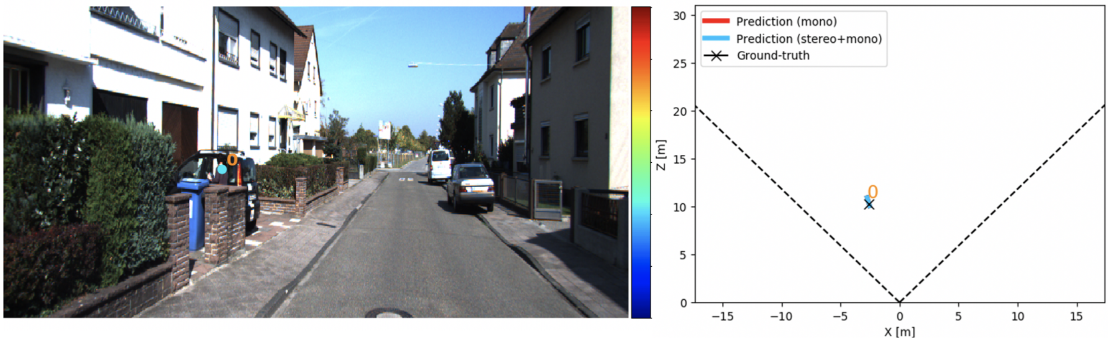


Figure 4.8 – A pedestrian covered by a low wall belongs to the category *hard*. MonStereo performs instance-based stereo matching and 3D localization with an error of 19 cm.

4.7.1 Close Instances

KITTI categories are defined based on occlusions, truncations and bounding box heights. Hence, *moderate* and *hard* instances often correspond to very close but partially occluded pedestrians. These types of instances deserve a great deal of attention and, in Figure 4.7, we show an example of it. A bike rack partially covers the pedestrian's legs in both images but MonStereo identifies a instance-based stereo matching and estimates depth with a localization error of 2 cm. Another example is shown in Figure 4.8, where the person is largely occluded by a low wall and belongs to the *hard* category. The person is 10 meters far and may soon cross the street. Early and accurate 3D localization is crucial for safety.

Challenges of Close Instances.

Humans occupy a 3D volume and estimating a single depth is not straightforward. Keypoints span over all the body and their disparities are not consistent for close instances. In Figure 4.9,

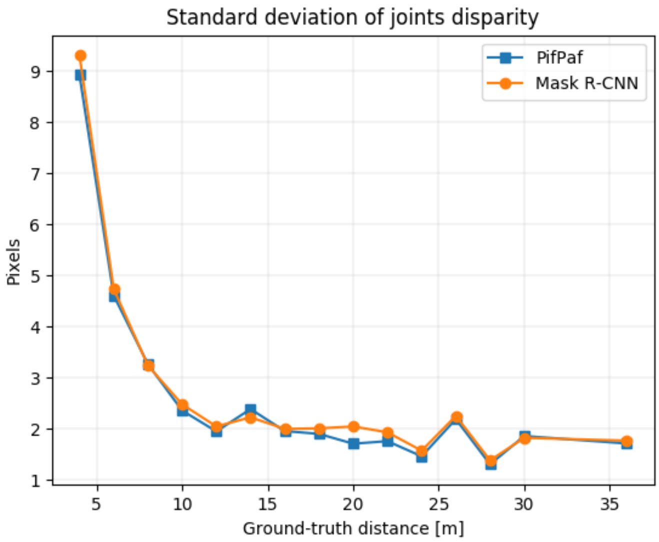


Figure 4.9 – Two sets of keypoint of each person in left-right images lead to 17 disparities. We analyze the standard deviation for keypoints obtained by two off-the-shelf pose detectors: Mask R-CNN [72] and OpenPifPaf [95]. The resulting performances are similar for the two pose detectors, highlighting that our method is agnostic to the choice of the detector. For very close instances, the standard deviation of keypoints disparity is high as humans are 3D entities and every body joint may be located at a different depth.

we analyze the standard deviation of keypoints disparity as a function of the ground-truth distance using two pose detectors: the top-down Mask R-CNN [72] and the bottom-up OpenPifPaf [95]. High variation of disparity for close instances can be concurrently caused by the 3D nature of humans and the quality of 2D keypoints. The two detectors lead to similar performances, highlighting the challenge of accurate 3D localization for close instances. Furthermore, the result shows that our method is agnostic to the choice of the pose detector.

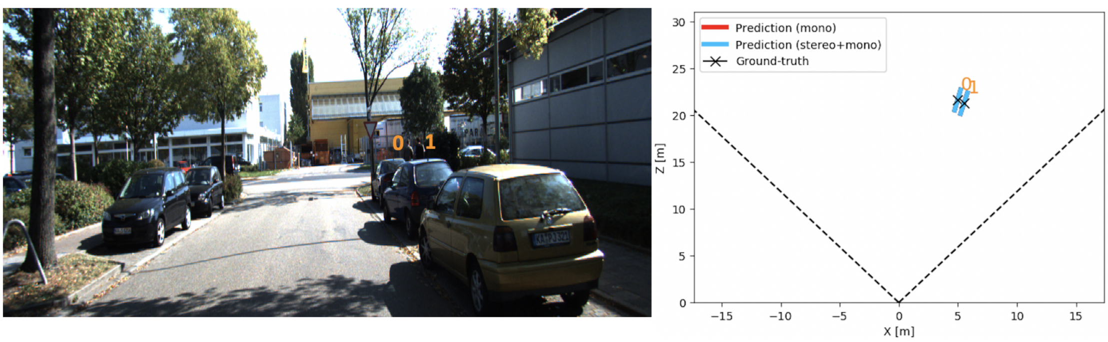


Figure 4.10 – Two far pedestrians heavily occluded by vehicles (*hard* category) are detected in both images and 3D localization is estimated with less than 5 cm error in both cases.

4.7.2 Far Instances

For far pedestrians, the standard deviation of keypoints disparities described in Figure 4.9 is greatly reduced, as a person only spans over few pixels. However, the depth error due to one pixel disparity grows quadratically with depth. In Figure 4.10, we show an example of two heavily occluded pedestrians 22 meters far, who are localized with only few centimeters of error. In Figure 4.11, we show a qualitative example of a "failure" as the instance 0 at 25 meters of distance is predicted with 69 cm of localization error. According to KITTI split, this instance belongs to the *easy* category. The performances for far instances are limited by the resolution of the camera: 69 cm of error corresponds to less than 0.5 pixel error in disparity estimation. On the contrary, all the people sitting at the café are closer and predicted with higher accuracy, but not evaluated by KITTI metrics as belonging to the category "person sitting".

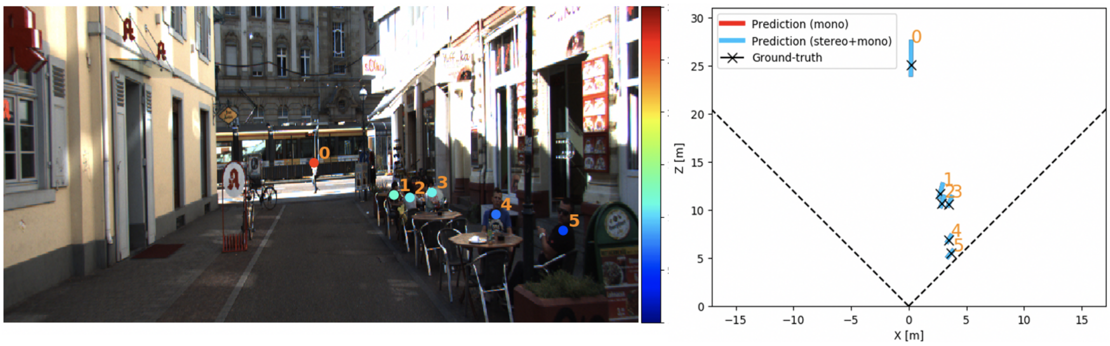


Figure 4.11 – A far pedestrian in the *easy* category is localized with a large error of 69 cm, while still included in the confidence interval. All the people sitting are localized with high accuracy, but not evaluated in KITTI metrics.

4.7.3 Details on Ground-truth Generation

KITTI dataset [64] does not include stereo matching information for instances in left and right images, but only depth of left instances. Hence, we extend ground-truth information to train our network for the instance-based stereo matching task. We detect a set of keypoints for each person using the off-the-shelf pose detector OpenPifPaf [95] and we compare the ground-truth depth with the one obtained through disparity estimation. For a given detection, the accuracy of every joint depends on many factors, such as occlusion. It is crucial to be able to detect and filter outlier keypoints that may affect our disparity calculation. Therefore, we adopt the following filters:

1. remove keypoints with confidence lower than a threshold;
2. remove outlier keypoints using Interquartile Range over the disparity estimation;
3. remove instances with large median vertical displacement.

Finally, we calculate the disparity as the median disparity of the remaining keypoints, we compare it with the ground-truth one and assign a binary label to the pair. We use an adaptive threshold which increases linearly with the depth, allowing for a larger error in case of far instances. This procedure does not involve the use of patches of images to analyze visual correspondences, being much faster and still very accurate. Our MonStereo reaches an accuracy of 98.2% on the validation set after being trained with binary cross-entropy loss.

4.8 Negative Results: Temporal Extension

A common alternative to stereo cameras is using multiple monocular frames over time. If the ego camera is moving, it is possible to calculate the distance of objects by their pixel disparity with just a monocular camera. While this approach, called structure from motion (SfM) [200], may in principle substitute stereo cameras, it is in practice more challenging. Stereo cameras are characterized by a constant disparity, while the SfM disparity is affected by the ego-motion of the camera and the target objects. In the autonomous driving scenarios, estimating the distance from SfM or stereo disparity is an equivalent problem only in the ideal case of a car at a constant speed and static pedestrians.

Problem Statement. In this section, we analyze the performances of our MonStereo by using two consecutive frames from a monocular camera as an alternative to stereo frames. KITTI dataset [64] does not include videos; hence, we use the recent nuScenes dataset [31] that provides video scenes of a car moving around the cities of Boston and Singapore. Ground-truths include the pedestrians' distance and their tracking ID over time with a frame rate of 2 Hz. We only work with images from the front-left camera, as the relative location of the camera with respect to the car affects the disparity. We consider as left-right pair of images the frames at time t and $t-1$, and our objective remains to estimate pedestrians' distance from monocular and disparity-based cues.

Architecture. The disparity is affected by the vehicle ego-motion on which the camera is mounted. Thus, we modify our network architecture of Figure 4.2 to include information from the CAN bus data, a vehicle bus over which information such as position, velocity, acceleration, steering, lights, battery are submitted. For simplicity, we choose to include as inputs to our network the vehicle speeds and steering angles at time t and $t-1$. We then upsample the signal with three consecutive fully connected layers with batch normalization [79] and a ReLU activation function and 128 features. The output features are then concatenated before the residual connections of our encoder.

Experiments. In Table 4.4, we show the ALE results of our experiments when using MonStereo with temporal frames and compare it with the monocular counterpart. We also evaluate the ISM task in different configurations to analyze the network performances in associating people between

Table 4.4 – Evaluating the performances of MonStereo with two monocular frames at 2 Hz. *Constant* conditions indicates we train and evaluate frames with the ego car at a constant speed and steering angle. *Stationary* conditions only includes frames where the car is stationary (while target pedestrians may still be moving). *True Pairs* uses ground-truth information to pair the instances between frames. In this case, the association task is superfluous, and the network could just use disparity-based cues instead of monocular ones. In each of these scenarios, we compare our MonStereo with a monocular baseline, where the network input is just the pedestrian pose at time t. Temporal-based results never outperform the monocular ones, showing that MonStereo mainly leverages monocular cues. Lower performances on the ISM task may also justify this.

Conditions	Instances	ISM	ALE (m)	ALE-Monocular
All	20K	87.1	1.39	1.36
Constant	2.3K	89.8	1.59	1.52
Stationary	27K	97.1	1.54	1.35
True Pairs	15K	-	1.25	1.22
Stereo Camera (KITTI)	5K	98.2	0.34	0.93

frames. The task is harder as the disparity is affected by the ego motion and the target pedestrians' one. Our results show an ISM accuracy of 87% compared to the stereo one of 97%. To simplify the problem, we also test a scenario where the car is stationary and another one in constant settings, namely:

1. $|v_t - v_{t-1}| < 0.2$,
2. $|s_t - s_{t-1}| < 10$,
3. $v_t > 3$,

where v is the vehicle speed in m/s and s the steering angle in degrees. The first two conditions approximates constant speed and steering between the two frames, while the third one assures the car is actually moving. Similarly to the other scenarios, the ALE error is comparable between MonStereo and the monocular counterpart, indicating that the network just uses monocular cues to estimate the distance. This is true even when we evaluate only *True Pairs* by using ground-truth information to input to the network pairs correctly associated between frames.

The results in Table 4.4 highlight the challenges of associating dynamic objects between temporal frames and using SfM to calculate their 3D localization. While our experiments on SfM are preliminary, they clearly show that using stereo cameras enables a privileged setup not easily replaceable by temporal information.

4.9 Conclusion

We have proposed a vision-based approach tailored for the long tail of 3D human localization. Our neural network implicitly learns to leverage monocular and/or stereo cues while communicating uncertainty driven by those cues. Our method goes beyond providing competitive results "on average" and shows reasonable estimates even for outliers. We hope this chapter will help direct the vision community's attention towards the long tail for autonomous driving and social robot applications, still an unchartered research territory.

In Chapter 3 and the current one, we demonstrated the effectiveness of semantic keypoints for the 3D pedestrian localization task. In the next chapter, we will study how to apply our methods for two real-world applications in the context of autonomous driving.

5 Autonomous Driving Applications of Pedestrian 3D Detection

This chapter is based on an applied research project in collaboration with the autonomous driving company Wayve.

5.1 Introduction

Our goal is to integrate AVs into our society, and for that to happen, they need to co-exist with humans in close proximity. The two previous chapters proposed vision-based methods to locate humans in 3D. This chapter explores two real-world applications of our methods to improve the interactions between self-driving cars and vulnerable road users.

One of the greatest challenges for self-driving cars is co-existing with dynamic agents. Among them, vulnerable road users present a separate set of challenges. First, the dynamics of pedestrians and cyclists are extremely complicated. People can often jaywalk or suddenly change direction without warning. Second, interactions between an autonomous vehicle and a pedestrian are not as frequent as interactions with other vehicles. Zebra crossings are mainly located in urban scenarios, and jaywalking may be considered an edge case behavior in most cities. Finally, the dynamics between a car and a pedestrian are affected by the surrounding environment. For example, at a zebra crossing without a traffic light, cars should always yield to pedestrians, but the presence of a traffic light affects all the dynamics. Unsurprisingly then, autonomous vehicles may fail to stop when a pedestrian is crossing the street or, vice-versa, they may engage in a too conservative behavior with unnecessary stops.

In the first application, we use our MonoLoco++ architecture, developed in Chapter 3, for creating pseudo-labels for 3D pedestrian detection. Low-cost data collection techniques are crucial to improving AV performances and their interactions with humans. For this purpose, we study the generalization capabilities of our model by training it on a publicly-available dataset and evaluating it on a custom-curated dataset in the city of London. In the second application, we propose a new approach to improve pedestrian awareness in AV systems. We first review the concept of AV 2.0 [71], and two patterns to improve the performances on pedestrian interventions for autonomous vehicles: data-centric and model-centric approaches. We then propose a new model architecture to add an inductive bias on the importance of pedestrians in the scene.

5.2 Pseudo-labels for 3D Pedestrian Detection

The autonomous driving field is driven by deep learning advancements [71], and this comes with the underlying requirement of collecting a massive amount of data. A common approach in the AV industry is to collect thousands of hours of data by driving with a fleet of vehicles and recording images and driver signals using the attached sensors. The collected data may directly be used to train end-to-end deep learning models, from sensory input to motion plan output. Nevertheless, training models with only throttle and steering labels make such models hard to interpret and debug. A possible solution is to modify the model to produce several intermediate outputs, such as 3D object detection or semantic segmentation. These outputs can be used for interpretability, safety verification, as well as to improve the overall performances [116, 37]. However, training models to produce intermediate representations usually require extra supervision with a large number of expensive labels. Labeling platforms are not feasible for labelling millions of images with human-interpretable representations such as semantic, geometry, and motion predictions. In addition, it is not clear in advance which labels are the most effective for debugging or improving performances.

In this section, we propose to unlock intermediate representations at a large scale by using pseudo-labels. We focus on the 3D object detection task, *i.e.*, 3D localization, orientation, and object dimensions for vulnerable road users. We compare our MonoLoco++ of Chapter 3 with FCOS3D [196], a new vision-based method that achieves state-of-the-art results in the nuScenes 3D detection challenge [31]. Our goal is to label millions of images recorded from vehicles driving in the city of London. To validate the quality of the pseudo-labels, we have created and labeled a small dataset of crowded traffic scenes in London, which we refer to as LDN. The dataset contains 3K instances for training and 3K instances for testing labeled with 3D bounding boxes.

Model Calibration. Our MonoLoco++ has the advantage of working with different monocular cameras without needing to be calibrated. Using semantic keypoints as an intermediate representation, we can back-project each keypoint into normalized image coordinates using the camera intrinsic matrix (more details in Section 3.4). This preprocessing step automatically scales the keypoints according to the focal length and image size. On the other side, end-to-end approaches, such as FCOS3D [196], are camera dependent. FCOS3D has been trained with specific camera parameters, and it does not adapt to different focal lengths. We correct FCOS3D predictions by calibrating them on our custom data to compensate for this. When using a camera with a different focal length, FCCOS predictions shift by a constant factor, and instances are estimated too close or too far from the camera. We correct the shifting using our custom data by minimizing the average signed error between the predictions and the ground truth.

Table 5.1 – Comparing FCOS3D [196] and our MonoLoco++ [23, 24] on the custom dataset LDN, which contains crowded traffic scenes in the city of London. FCOS3D has been trained on the full nuScenes dataset (nS-F) [31] that contains 1.4M training instances, while MonoLoco++ has been trained on the nuScenes teaser (nS-T). Our MonoLoco++ performs better in the average localization error (ALE) and average orientation error (AOE) even when trained on 100 times fewer instances. In addition, the use of semantic keypoints allows adapting to different cameras easily. In the case of FCOS3D, using the correct intrinsic matrix as input is not sufficient, and we had to scale the results with a corrective factor (indicated with *).

Method	Training Dataset (instances)	ALE (m) ↓ [Recall (%) ↑]				AOE (°) ↓
		Easy	*Mod.*	*Hard*	*All*	
FCOS3D	nS-F (1.4M)	1.58 [72%]	2.21 [57%]	3.46 [22%]	2.57 [36%]	71.0
FCOS3D*	nS-F (1.4M)	0.85 [72%]	1.31 [57%]	3.30 [22%]	2.01 [36%]	71.0
MonoLoco++	nS-T (8K)	0.74 [84%]	1.44 [61%]	2.69 [12%]	1.54 [31%]	30.7
MonoLoco++	LDN (3K)	0.70 [84%]	1.28 [61%]	**2.45** [12%]	1.40 [31%]	30.6
MonoLoco++	nS-T+LDN (11K)	**0.68** [84%]	**1.23** [61%]	**2.45** [12%]	**1.38** [31%]	**26.5**

Evaluation. We use the average localization error (ALE) and the average orientation error (AOE) metrics for evaluation. We compare FCOS3D [196] trained on the nuScenes dataset [31] with 1.4M instances with our MonoLoco++ trained on the teaser version of nuScenes that contains 100 times less instances. We also compare with versions of MonoLoco++ trained on the LDN training set only and both datasets. As shown in Table 5.1, our method achieves better results on all metrics. Results only marginally improve when training on both nuScenes and LDN, indicating good domain adaptation properties. The use of semantic keypoints as intermediate representations makes the architecture camera independent and removes background artifacts that encourage overfitting. We also analyze in more detail the ALE metric as a function of the ground-truth distance in Figure 5.1. We compare our results with the upper bound of performances for monocular methods, represented by the *task error* of Equation 3.2. Up to 20 meters, the error grows linearly due to the ambiguity of perspective projection. From 20 meters on, the noise in the predicted keypoints becomes a dominant source of error. A qualitative example is shown in Figure 5.2, representing a crowded intersection in London from a vehicle perspective.

These results highlight that our MonoLoco++ can generalize well to unseen scenarios. In the next application, we are going to integrate our architecture into a more complex end-to-end motion planner to improve AVs awareness of vulnerable road users.

5.3 Pedestrian Awareness for AV2.0 autonomous driving systems

In this section, we review the concept of AVs 2.0 [71] and we propose a new model architecture to add an inductive bias on the importance of pedestrians in the scene. In our study conducted in the city of London, autonomous vehicles (AVs) often slow down in the presence of pedestrians but they do not always stop when they should. The vehicles analyzed in this study deploy end-to-end models trained with the control signals from expert drivers, without any direct supervision on

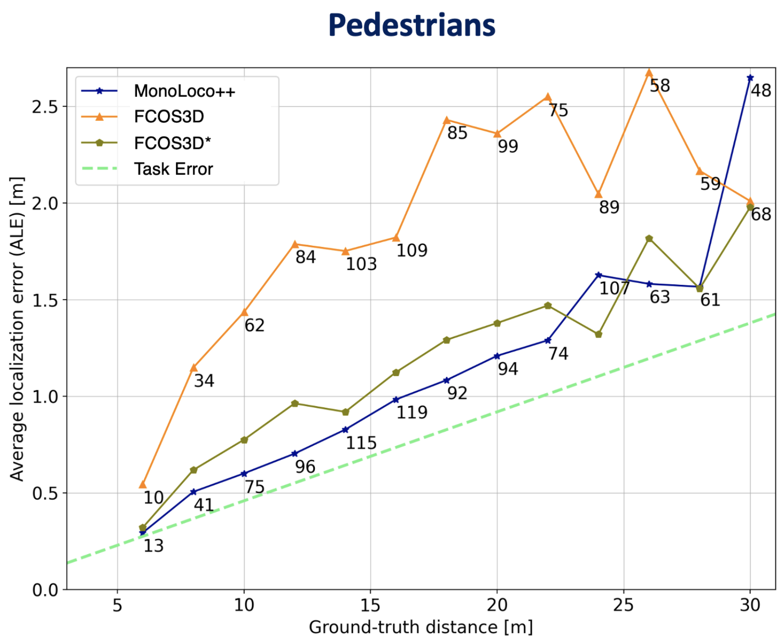


Figure 5.1 – Average Localization Error (ALE) as a function of distance for FCOS3D [196] and MonoLoco. FCOS3D* is corrected to account for a different focal length. MonoLoco++ achieves robust performance while detecting a comparable number of instances in each cluster (numbers included). The task error represents an upper bound of performances for monocular methods.

pedestrian location or behaviour. Our hypothesis is that the model is averaging its response between stopping and navigating at a constant speed, as it is not able to correctly capture pedestrians' dynamics (*e.g.*, being close to a zebra crossing is not a sufficient condition to infer that the pedestrian is going to cross).

To test our hypothesis, we use as a case study one of the development models created by the autonomous driving company Wayve, which we refer to as W1. It is a fully learned end-to-end motion planner trained with imitation learning. It receives as input a single image, the GPS location, the current speed, a coarse route map, and outputs the vehicle trajectory in the following seconds [117]. Our goal is to improve the intervention rate of autonomous vehicles due to pedestrian interactions, either by managing data or by improving the current W1 model. We test both approaches by (i) leaving the W1 architecture unaltered but changing the training data curriculum and (ii) modifying the W1 model to inject knowledge about pedestrians. For the latter option, we build on the MonoLoco++ architecture developed in Chapter 3 to convert the 3D location, orientation, and uncertainty estimation into a bird's eye view occupancy map. We then upsample and fuse this occupancy map as additional features into the main model without adding any supervision or additional loss. The overall model is still trained end-to-end with imitation learning but we are nudging it to pay attention to pedestrian dynamics through the fusion of additional features. The use of an external, pre-trained network to detect vulnerable road users has two advantages. First, it increases the interpretability of the architecture. In case of a failure mode, it is possible to debug whether the pedestrian or cyclist has been detected by the external

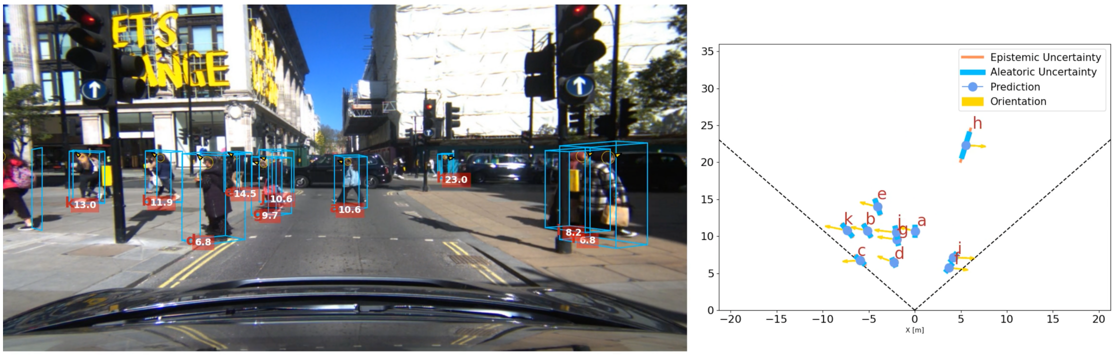


Figure 5.2 – Example of a crowded scene in the city of London from a car perspective. Our network estimates the 3D localization, orientation, dimensions, and uncertainty of each vulnerable road user.

network. Second, it increases redundancy against out-of-distribution samples as the network has been trained on different datasets.

5.3.1 AV 2.0 Framework

The navigation task from a robotic perspective has been traditionally tackled with the sense-plan-act paradigm [179], which is composed of two main pillars: perception and planning. Dividing the task into sequential steps is attractive for its interpretability and because it enables parallel progress on each sub-task [115]. However, this pipeline imposes fundamental limits on the autonomous driving development. It often leads to compound errors by neglecting the interactions between the stages and its sequential nature does not foster agility and speed [115]. Hawke *et al.* [71] from the self-driving cars company Wayve argue that with the traditional sense-plan-act paradigm, referred to as AV 1.0, AI could never be scaled rapidly to new environments or vehicle types. In their view, only a complete rethink of these decision-making systems could move the AV research forward. The main argument is that the perception and planning pillars (with the exception of behavior prediction and planning) are already mature enough and further development would not unlock structural improvements in autonomous driving. The authors claim that the main issue the AV industry is facing is the ability for AVs to generalize to unseen scenarios, and that only a data-driven, end-to-end pipeline could solve it. The AV 2.0 framework consists in solving driving with data, by creating an end-to-end architecture that implicitly combines the perception and planning pillars. As of 2021, the autonomous driving community is adhering more and more to the AV 2.0 trend, sharing the concept of a fully-differentiable AV stack. Among them, Lyft Level 5 is focusing on the importance of data-driven simulation [88], and Uber ATG is working on end-to-end models with intermediate interpretable representations [37].

The AV 2.0 framework can be seen as the combination of two approaches: a data-centric approach and model-centric one, as data collection is as critical as model improvements towards scalability issues [88, 71]. We review the two approaches using as case-study the fully learned end-to-end motion planner (W1) developed by the self-driving cars company Wayve. In the experiments described here, we have focused for simplicity and speed of iteration on a variant of the model that only uses monocular input and no temporal context, nor any 3D information in the form of LiDAR point clouds or stereo cameras. The output is a bird's eye view cost map indicating the likelihood of the trajectories in the next few seconds. The most feasible trajectory is then passed to the actuator module to be converted into acceleration and steering signals. The architecture is inspired by [144, 76, 117] and the core idea consists of first lifting the features extracted from an image into the 3D space and then projecting them into the bird's eye view space. At this stage, the features from the image are concatenated with the features extracted from the current speed and the coarse road map before being processed in the planning module. All the components are trained end-to-end without any independent hand-engineering of the different modules. An overview of the W1 components is shown in Figure 5.4.

Data-centric Approach. It consists of improving the performances by focusing on the data curriculum aspect of the training pipeline. W1 has been trained to estimate the best future trajectory at any time step, given just a monocular image as input. In the context of imitation learning, ground-truth trajectories are obtained by recording two types of data:

1. **Expert data**: standard data collected recorded speed, steering, and acceleration labels of expert drivers during navigation.
2. **Corrective action (CA) data**: intervention data collected while disengaging from autonomous mode. The expert driver intervenes to correct the trajectory of the W1 model when necessary. Speed, steering, and acceleration labels during the intervention are recorded.

To improve pedestrian awareness through data, we modify the data curriculum to upsample both expert data in proximity of zebra crossing and corrective action data due to a pedestrian intervention. This approach excels for its simplicity. However, CA data may be recorded during an emergency response and should not represent the majority of the training data. Also, the data-centric approach requires a careful analysis to validate whether a model trained with a different data distribution does not worsen its performances on different behavioural competencies. In this chapter, we mainly focus on the model-centric approach and use the data-centric one as a baseline.

Model-centric Approach. It consists of modifying the model architecture and its parameters without altering the data distribution. We first review two common techniques and their limitations to improve the vehicles' intervention rate in the presence of pedestrians, and then describe the proposed approach. A widely-used method is multi-task learning [215], which consists of training a neural network with a common encoder and two or more separate decoders. In this case, one decoder is used for trajectory prediction, and the additional one is used for 3D object detection or dense depth estimation. We consider this approach *soft* as the additional loss does not enforce

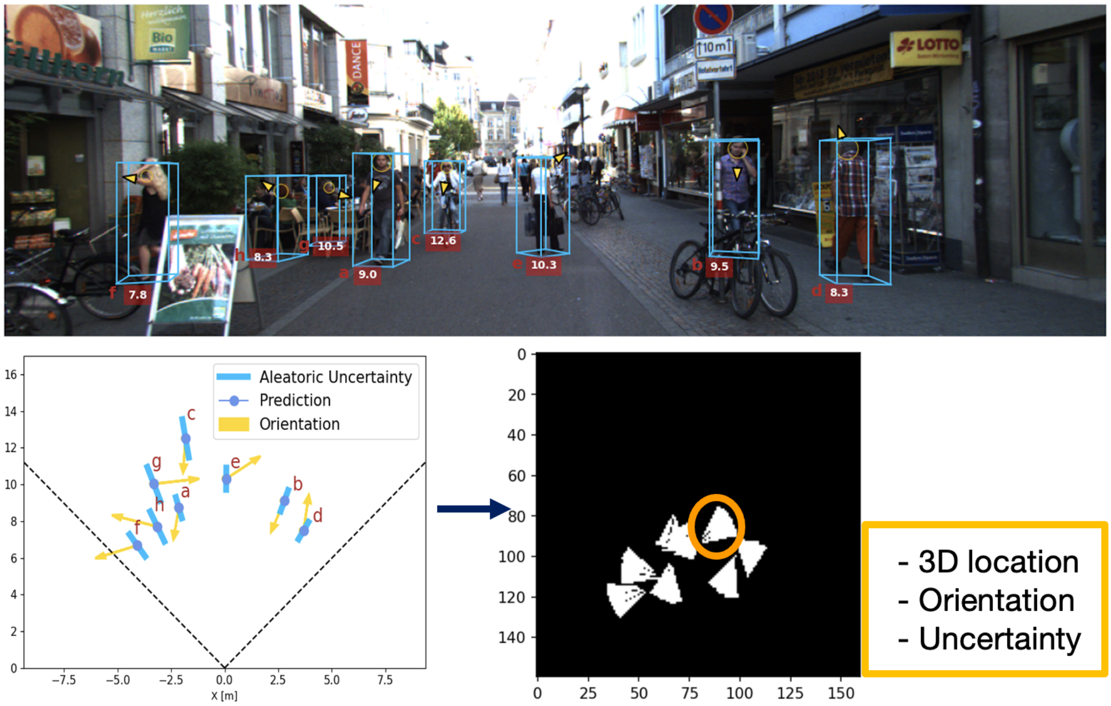


Figure 5.3 – Visual illustration of the occupancy map generated by MonoLoco++. Each pedestrian is detected in 3D, projected into a discrete bird's eye view map, and represented as a cone. The origin of the cone corresponds to the predicted 3D location, its width to the predicted aleatoric uncertainty, and its orientation to the body orientation of the person.

the model to slow down in the presence of pedestrians, but only to detect them. As pointed out by Standley *et al*. [185], multi-task learning does not guarantee improved performances, and smaller independent networks often achieve better performances. At the same time, this approach requires explicit supervision on 3D object detection labels, which may not be available at training time. A second technique is the use of collision losses [17]. It consists of penalizing trajectories that collide with vulnerable road users. This is a *hard* approach as it enforces the expected behaviour of the model. At the same time, it is limited in its scope, as a trajectory is not necessarily correct only when it avoids collisions with pedestrians. For example, AVs also need to yield to pedestrians, slowing down as soon as they understand the pedestrian intention even if a collision is not foreseen yet. Our proposed approach lies at the intersection of a traditional *soft* approach such as multi-task learning, and a *hard* approach that explicitly enforces constraints.

5.3.2 Pedestrian-aware Occupancy Map

We inject knowledge into the model regarding the 3D location and orientation of vulnerable road users using the pre-trained network MonoLoco++ [24]. We first discretize the network output into a binary occupancy map of 32 meters of height and width, with a grid size of 160. As shown in Figure 5.3, we transform each detection into a cone with the origin corresponding to

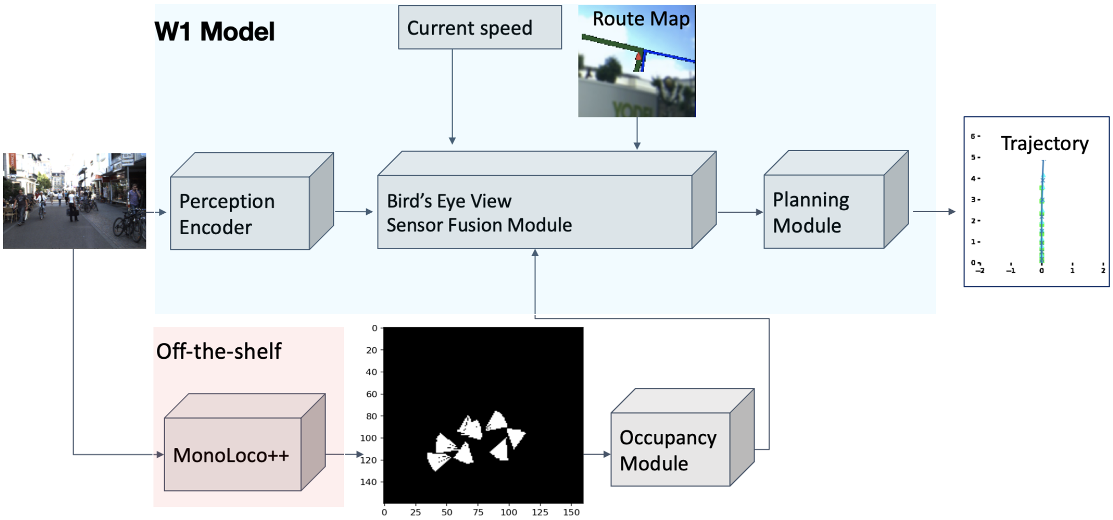


Figure 5.4 – Overall architecture of an end-to-end model for autonomous driving, which we refer to as W1. The inputs are a monocular image, the current speed, and a coarse route map, and the output is the planned trajectory in the next few seconds. The main blocks are a perception encoder, a sensor fusion, and planning modules. We inject knowledge about pedestrians by combining the W1 intermediate features with the output of our MonoLoco++ architecture. The occupancy module upsamples the occupancy map from a two-dimensional 1-channel feature map into 32-channels features with the same spatial resolution.

the predicted 3D location. The width of the cone is given by the predicted aleatoric uncertainty of the network, and its orientation represents the predicted body orientation of the person. We treat this binary occupancy map as a 1-channel feature map, and we pass it into four blocks of convolutional layers, each one followed by batch-norm [79] and ReLU activation function [131]. This occupancy module transforms the occupancy map into 32-channels features with the same spatial resolution of 160. The features are then concatenated with the intermediate features of the W1 model, which have been selected in accordance with the W1 model structure.

In the W1 model, after the perception encoder, the features are first lifted to 3D and then projected back to a bird's eye view canvas of 32m x 32m and the same grid resolution of 160 [76]. At this stage, the W1 features are concatenated with features coming from the current speed and the coarse road map. Then, we extend the current W1 structure to fuse features coming from our occupancy map. The model is then trained end-to-end with the exception of MonoLoco++ module that is considered as an off-the-shelf network.

5.3.3 Experiments

Testing a new feature in an AV model is a daunting task. The real world is multi-modal and labels are usually not available. In addition, we expect a model to improve on a given task, for example pedestrian awareness, while maintaining consistent results in the whole operational driving domain distribution (right turns, cycle lanes, etc). For the sake of this study, we only

Figure 5.5 – Qualitative example of an out-of-distribution sample. A pedestrian is bending to pick up an object. Our off-the-shelf network recognizes the pedestrian and estimates a reasonable 3D localization and orientation

focus on evaluating the model performances on pedestrian interventions in "shadow mode". It is an open-loop, offline protocol that registers how the car would have acted if the computer was in control [65]. It is effective to evaluate independent frames but it cannot be used for temporal evaluation, as the predicted action at a given time step will not actually affect the future vehicle behaviour.

Quantitative Results. We create a test set by collecting CA data due to pedestrian interventions. It consists of frames and labels where the expert driver intervened because the vehicle in autonomous mode did not give way to pedestrians. In these cases, the AV was either driving too fast or not yielding, and the expert driver had to slow down or stop. Hence, we evaluate the speed of our new models in shadow mode at the time of intervention, as shown in Table 5.2 as well as with the histograms in Figure 5.6. We compare a baseline W1 model, a data-centric approach (DCA), and a model-centric approach (MCA). The DCA shares the same architecture of the W1 model but it is trained by upsampling expert data in proximity of zebra crossing and corrective action data due to a pedestrian intervention. The MCA is the modified version of the W1 model that includes the occupancy map as depicted in Figure 5.4. The results are shown in Table 5.1 as well as in Figure 5.2. The best performing approach is the DCA, while the MCA obtains approximately the same results as the baseline. Our hypothesis is that the occupancy map layers did not learn a meaningful representation of vulnerable road users due to the sparsity of pedestrians in the training set. In fact, in the training set for the large majority of frames, the occupancy map remains blank. To test our hypothesis, we substitute the actual occupancy map

Table 5.2 – Average speed at the time of intervention with a baseline method (W1), a data-centric approach (DCA), a model-centric one (MCA), and a combined approach (DCA + MCA). The lower the speed, the better, as we collected only the frames where the expert driver intervened to slow down due to a pedestrian crossing.

Method	Average Speed [m] ↓
Baseline (W1 Model)	13.6
DCA	**10.4**
MCA	13.5
DCA + MCA	11.2

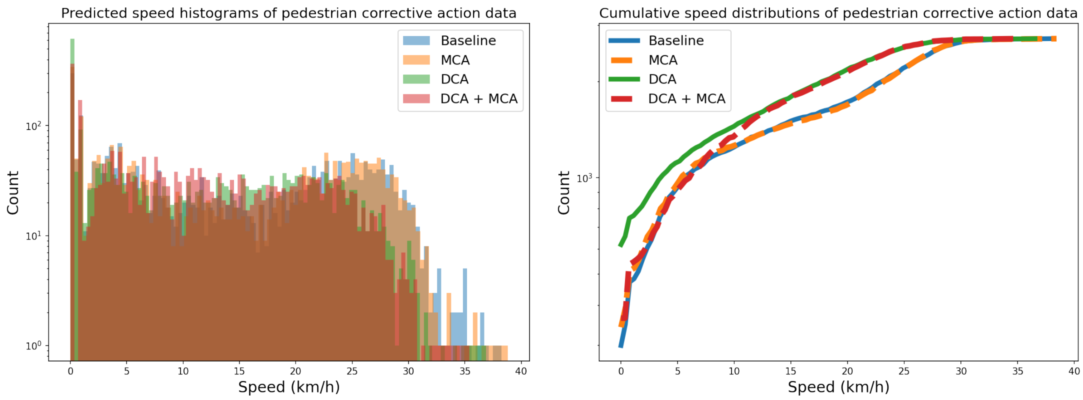


Figure 5.6 – Predictive and cumulative speed distributions at the time of intervention with a baseline method, a data-centric approach (DCA), a model-centric one (MCA), and a combined approach (DCA + MCA). The most effective approach to reduce speed when needed is the data-centric one.

with an empty one and notice that the model predictions remained the same, meaning that the model has learned to ignore the occupancy map layers.

These results speak in favour of the data-centric paradigm for real-world testing at scale, as this paradigm is not only more effective in these scenarios but also simpler to implement. This experiment demonstrates once again how large-scale, low-cost data collections [80] and data curriculum [71] are critical solutions towards scalability issues.

Qualitative Results. While using a DCA improves overall performances, the use of MCA with an external network can add interpretability and, potentially, robustness towards out-of-distribution samples. A qualitative example is shown in Figure 5.5, where MonoLoco++ detects a pedestrian that is bending over to pick up an object. This image is recorded in the city of London from a camera mounted on a vehicle. In the context of autonomous driving scenarios, the pedestrian may be considered as an outlier due to the unnatural pose. On the other side, pose estimation datasets such as COCO [111] include a large variety of poses and this may justify why the pedestrian is correctly detected by the pose estimation algorithm OpenPifPaf [95].

Model Deployment and Run Time. Deploying a deep-learning model in a vehicle presents additional challenges, often overlooked when developing research models. Arguably, the main one is serializing the model to embed it into high-performance production environments such as C++. In our case, we use TorchScript [141] to convert a PyTorch model into an intermediate representation that can run outside of Python, via LibTorch, a C++ native module. A scripted model gains flexibility and execution performances at the price of extra development effort. For a model to be converted, its initialization and forward pass can only be constructed using torch tensors or Python constants. This constraint is particularly challenging in our case, where both the OpenPifPaf decoder [94] (not differentiable) and the keypoints' preprocessing steps were not originally developed in PyTorch. With a serialized model, we obtain an average inference time of 128 ms with a raw image of 960x600 resolution using a machine with a single NVIDIA GeForce GTX 1080 Ti.

5.4 Conclusion

In this chapter, we have studied how to apply our 3D pedestrian detector MonoLoco++ for real-world applications in the autonomous driving field. First, we have created a pipeline to obtain low-cost pseudo-labels. We have adapted our MonoLoco++ to produce 3D pedestrian detection labels and evaluated it on a new custom dataset, hence studying its generalization properties. Second, we have studied two approaches to improve pedestrian awareness in end-to-end models: data-centric and model-centric. We have worked with a real-world development model for autonomous vehicles in the city of London. Specifically, we have experimented with (a) improving the data curriculum and (b) developing a new model-centric approach to highlight the importance of pedestrian features. Our experiments have demonstrated that, for this application, the data-centric approach was the most effective one to improve the network performances.

In general, we encourage the vision community to direct their attention towards data-centric approaches. Often, researchers focus solely mainly on model-centric approaches, training and comparing different models on the same benchmark to carry out fair and reproducible experiments but there is plenty of unexplored territory and scope for further improvements with both approaches. As this case study has highlighted, data-centric approaches, such as data curriculum, may play a critical factor in improving performances for real-world applications.

In Chapters 3, 4, and the current one, we have analyzed the effectiveness of semantic keypoints in locating humans in the 3D world using cost-effective vision-based sensors. However, autonomous agents cannot safely navigate around pedestrians by just estimating their 3D distance; they must understand humans' actions and infer their behavior. Luckily, humans communicate intentions through body language, *e.g.*, a look over the shoulder to determine whether it is safe to stop and turn. The next chapter will focus on perceiving an essential communicative action: eye contact. In urban or crowded environments, humans rely on eye contact for fast and efficient communication with nearby people. Autonomous agents also need to detect eye contact to interact with pedestrians and navigate around them. In the following chapters, we will also be expanding our focus to different types of actions, studying atomic actions, such as walking or sitting, and communicative action, such as talking in groups.

6 Detecting Pedestrians Attention: Human-Robot Eye Contact in the Wild

This chapter is based on the article:

Younes Belkada⋆, Lorenzo Bertoni⋆, Romain Caristan, Taylor Mordan, and Alexandre Alahi, *Do Pedestrians Pay Attention? Eye Contact Detection in the Wild* (Under review)

and on the extended abstract:

Weijiang Xiong, Lorenzo Bertoni, Taylor Mordan, and Alexandre Alahi, *Simple Yet Effective Action Recognition for Autonomous Driving*, (Under review)

6.1 Introduction

In the previous chapters, we have focused on locating humans in 3D. However, AVs also need to understand humans' actions and their future intentions to move naturally around humans and avoid collisions [155, 158]. Intentions are usually communicated with body language, and in this chapter, we focus on detecting eye contact between humans and a target robot, such as an AV or a delivery robot. When walking or driving, people use eye contact to pay attention to their environments, acknowledge the presence of others, and communicate intentions. Hence, AVs need to understand whether a pedestrian intends to cross the street in front of the vehicle [156, 191]. Similarly, smaller robots moving in crowds should be capable of detecting whether pedestrians have noticed them and are more likely to avoid them actively [91, 92]. Finally, even in smart cities, detecting eye contact between a human and a robot can be helpful in better understanding pedestrians' future behaviors [13] by identifying where their attentions go or what public signs they are looking at.

This chapter focuses on detecting whether humans are looking at the camera, hence performing eye contact with a target robot on which the camera is mounted. This communicative action is informative on whether pedestrians are paying attention to the incoming AV, ultimately affecting

⋆ denotes equal contributions

Figure 6.1 – Typical scene for eye contact detection *in the wild*, where pedestrians might be far from the camera and heavily occluded. Our method estimates, from predicted body poses, whether people are paying attention (showed in green) to the ego camera through eye contact, or are distracted (showed in red). This information can then help to better forecast their behaviors and to reduce the risk of collision with a self-driving agent.[I]

the vehicle trajectory. For clarity, we will refer to this task as *eye contact detection* in the reminder of this thesis.

Although humans make eye contact with each other at all times, detecting this communicative action in the wild, *i.e.*, with no constraint on the environment such as exemplified in Figure 6.1, presents a few challenges. First, the action can be quick and subtle, happening with small head movements lasting as short as a few milliseconds. Because of this small window, both spatially and temporally, the detection is hard and can easily be affected by environmental conditions, such as lighting or distances of pedestrians. Furthermore, eye contact has received little attention, and few datasets have been annotated with it [156, 154], when compared to more popular vision tasks such as object detection [54] or 2D pose estimation [111]. These reasons make it more difficult for autonomous systems to detect eye contact effectively and generalize to new environments.

In order to mitigate these issues, we propose to detect eye contact from high-level semantic keypoints, as displayed in Figure 6.1. Although one could expect images to be a crucial input representation for eye contact detection, we show that we can use keypoints as input to escape the image domain, and process them with a simple, yet effective neural architecture. For this, we first rely on a pose estimation step, which extracts semantic keypoints for all pedestrians in an

[I]Image under license CC-0, https://jooinn.com/images1280_/people-walking-on-pedestrian-lane-during-daytime.jpg.

image, using the off-the-shelf pose detector OpenPifPaf [95]. Using pose features as input rather than images presents several advantages. As pose needs less resolution than gaze, while also being annotated on more diverse datasets, it should be less affected by noise from environmental conditions, and predictions should generalize better to different scenarios and environments. Poses are also much less dimensional than images, and do not require as much network capacity to be processed properly. This allows the use of lighter networks, which should help prevent overfitting on the few scenarios annotated. Finally, by leveraging these high-level features, we remove background information and reduce effects from changes in image statistics, allowing our model to focus solely on eye contact detection.

Since there are not many datasets covering a large variety of scenarios for eye contact, and these usually include a limited number of pedestrians [156, 154], we argue that if a model is to be trained on them and deployed in the real world, it then must be particularly able to generalize well to new, uncontrolled conditions. We suggest evaluating this through cross-dataset generalization, and we annotate three common autonomous driving datasets with this new task, namely KITTI [64], nuScenes [31], and JRDB [121], to diversify the scenarios involving eye contact. When evaluating our models, we show that using semantic keypoints leads to models generalizing better to various datasets and scenarios. We publicly release the annotations as a new dataset, which we refer to as LOOK[II], as well as the source code[III], towards an open science mission.

Driven by the effectiveness of semantic keypoints for eye contact detection, we expand our study at the end of the chapter and focus on various action recognition tasks in the context of transportation applications. We use our approach to determine which tasks keypoints are effective representations for. We first validate our approach on the TCG [201] dataset, showing that a simple method can achieve better results than temporal baselines using LSTMs [74], and comparable results with complex attention-based graph convolutional networks. Then, we compare our approach on various action recognition tasks on TITAN [119], a new dataset for autonomous driving. We show that on atomic actions, such as *walking* or *sitting*, our keypoint-based approach outperforms image-based methods, validating the effectiveness of human poses as intermediate representations for action recognition tasks in transportation applications.

To summarize, our contributions are as follows:

- We propose a deep learning model leveraging semantic keypoints, specially adapted to the challenges of human-robot eye contact detection. We further adapt our model for various action recognition tasks in the context of transportation applications.
- We publicly release LOOK, a diverse, large-scale dataset for human-robot eye contact detection in the wild, with numerous unique pedestrians and a focus on generalization across domains and scenarios, by annotating three common autonomous driving datasets;

[II] Dataset: https://looking-vita-epfl.github.io
[III] Source code: https://github.com/vita-epfl/looking

- We suggest an evaluation protocol for eye contact detection with real-world generalization in mind, and show that our approach yields state-of-the-art results and strong generalization compared to image-based methods.

6.2 Related Work

6.2.1 General Eye Contact

Gaze estimation has received a lot of attention from the Computer Vision community, as a simpler alternative to eye tracking. As for most other tasks, all leading approaches now rely on Deep Learning to get state-of-the-art results on the various benchmarks [214]. In this paper, we focus on eye contact detection, which can be considered as a special case of gaze estimation. There are multiple works that tackle this problem, *e.g.*, Smith et al. [181] focus on gaze locking from eyes' visual appearances by masking out their surroundings, Park et al. [140] transform single eye images into simplified pictorial representations to regress the angle of the gaze. Some others focus also on real-time inference. Fischer et al. [58] use a cascade of networks to localize heads and face landmarks, to align them to a predefined normalized face image for extracting eye patches, then compute gaze with a deep network. Rowntree et al. [165] also use two networks for head detection and gaze estimation in order to speed up the overall pipeline.

The major issue with these works is that the benchmarks and the methods are not applied to *in-the-wild* applications for autonomous vehicles, where the resolution of the pedestrian is low. They usually focus on situations where people are rather close to the camera (*e.g.*, inside a vehicle, in front of a computer), where the heads occupy larger regions in the images, and with simple or plain backgrounds, sometimes in controlled setups. On the other hand, we focus on eye contact detection *in the wild*, where there is no prior constraint on the type of environment pedestrians are in.

6.2.2 Eye Contact between Pedestrians and Vehicles

From a driver's perspective, detecting eye contact is an important cue that indicates pedestrians' awareness of the traffic and future crossing intentions. However, few datasets have annotated this action. JAAD [156] and PIE [154] are two such datasets, both focusing on pedestrians likely to cross the road in front of vehicles. They therefore allow learning and evaluating eye contact directly from images. Rasouli et al. [156] use an AlexNet [96] to classify cropped images of pedestrians' heads but require bounding boxes to be given. Varytimidis et al. [191] have a similar approach where they use an SVM to classify features from a convolutional network applied to head crops, and then process their predictions with contextual information. Mordan et al. [126] jointly detect pedestrians and eye contact, along with other attributes, in a single network forward pass using multi-task fields.

In the context of pedestrian crossings, multiple works (in addition to the previous ones) use eye contact as an intermediate feature to better predict pedestrians' future behaviors. Kooij et al. [91] estimate head orientation as a cue for pedestrians' situational awareness, and use it with other indicators to predict their paths around the road with a Dynamic Bayesian Network. Other approaches use a similar strategy for pedestrian awareness, *e.g.*, Hariyono et al. [70] for estimating the risk of collision, Kwak et al. [103] for prediction pedestrian intention at night time.

Eye contact detection is directly related to whether pedestrians pay attention to the incoming traffic. In practice, detecting that they do not pay attention is as important. One of the main reasons for that is the use of a phone that draws their attention away from the road. Identifying phone-related activities is therefore an insightful cue to detect. Rangesh et al. [153] show the practical importance of having gaze annotations, both for eye contact between drivers and pedestrians, or phone-related distractions. Going further to recognize actions implying a phone, Saenz et al. [169] use a two-branch convolutional network to predict distracted behaviors due to phone usage from stereo image pairs.

While detecting phone-related activities or pedestrian intentions are crucial tasks, eye-contact detection remains an essential channel of communication. Pedestrians may have the intention to cross but have they seen the upcoming car they should yield to? Contrarily, people may hold the phone but still pay attention to the upcoming traffic.

6.3 LOOK Dataset

We argue that eye contact detection is a crucial yet under-explored task to enable autonomous agents to safely navigate around pedestrians. To promote research in this area, we show that the current datasets are not sufficiently diverse for data-driven methods, and we create a new large-scale dataset for eye contact detection in the wild.

6.3.1 Existing Datasets

To the best of our knowledge, only two datasets contain annotations for the eye contact detection task: JAAD dataset [157], and PIE dataset [154]. JAAD consists of 390K instances of pedestrians labeled with bounding boxes and behaviour annotation, of which 17K instances have been labeled as looking at the driver (*i.e.*, at the camera in the car). The dataset is large in size but limited in diversity. It is made of 346 video clips of 5-10 seconds recorded with an on-board camera at 30fps in North America and Europe. Thus, many frames show the same people in similar scenes, and the number of unique pedestrians looking at the camera is 686.

PIE [154] is also a recent dataset for pedestrian intention estimation. It shares many of the characteristics of its predecessor JAAD [157]. It is recorded using an on-board camera at 30fps and consists of continuous footage of 6 hours in downtown Toronto, Canada. Out of 700K annotated pedestrian instances, there are 1,842 unique pedestrians, of which less than 180 are looking at the camera.

Table 6.1 – Dataset statistics. *Frames* is the total number of frames in the datasets. *Pedestrians* indicates the number of unique pedestrians, while *Instances* counts the number of occurrences of pedestrians in all frames. In brackets, we mention the percentage of instances that are looking at the camera. JAAD [157] and PIE [154] datasets include a very large number of instances but in comparison a very small number of different pedestrians. In contrast, our LOOK dataset includes in total 7,944 unique pedestrians from three continents, enabling exhaustive studies on cross-dataset generalization.

Dataset	Frames	Instances [% looking]	Pedestrians
JAAD [157]	82K	133K [18%]	686
PIE [154]	909K	739K [9%]	1,842
Our LOOK-KITTI [64]	1,391	4.630 [17%]	425
Our LOOK-JRDB [121]	9,441	39K [18%]	399
Our LOOK-nuScenes [31]	2,216	13K [9%]	7,100
Our LOOK	13K	57K [16%]	**7,944**

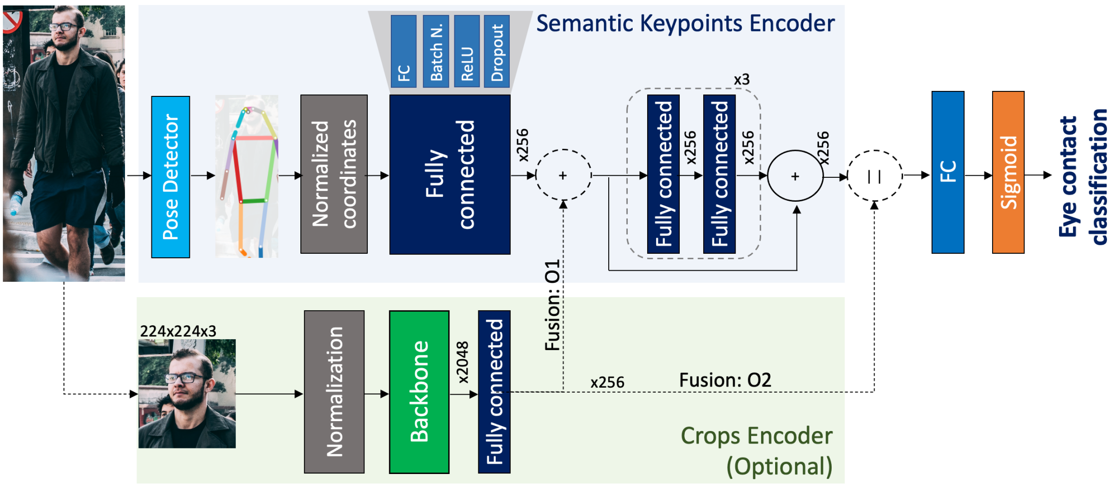


Figure 6.2 – Modular architecture: the input of our keypoint-based model is the set of 2D keypoints extracted from a raw image, and the output is the binary flag indicating whether a person is looking at the camera. A *Fully connected* block outputs 256 features and includes a fully connected layer (FC), a Batch Normalization layer (BN) [79], a ReLU activation function, and dropout [183]. Optionally, the features obtained from the semantic keypoints are concatenated with the features obtained from the head crops. We experiment with two types of fusions in the early (O1) or late (O2) layers, and with different convolutional architectures, such as ResNet-18 [73] or ResNeXt-50 [207] as backbones for the crop-based module.

6.3.2 Benchmark Selection

We have built a new large-scale dataset for eye contact detection in the wild by selecting publicly available images from three existing datasets: KITTI [64], nuScenes [31] and JRDB [121]. The first two are autonomous driving datasets and are made of images taken from a driver perspective. The latter one consists of videos taken from a small robot moving in crowded spaces inside Stanford University campus. In total, we have labeled 57K images from four different cities (Boston, Singapore, Tübingen, Palo Alto) in three continents. We aim for diversity, selecting pedestrians areas [64], crowded images from six cameras around the car [31], and indoor environments from a robot perspective [121]. In total we have labeled around 8,000 unique pedestrians, making it the most diverse dataset for eye contact detection in the wild. Examples from the LOOK dataset are shown in Figures 6.4a, 6.4b and 6.4d.

We provide, together with the dataset annotation, the training and testing splits. We make sure that splits do not contain overlapping scenes and that the test set is sufficiently diverse, including 22% of the total number of unique pedestrians over 15% of the frames.

6.3.3 Annotation Pipeline

Our LOOK dataset has been annotated using the Amazon Mechanical Turk (AMT) platform. Each image has been annotated by four workers, which had the options to select whether a person was looking at the camera, somewhere else, or none of the two in case of ambiguity. We then include in the dataset only the labels with a consensus of at least three out of four annotators. This threshold has been selected by reviewing edge cases where not all the workers agree on a selected instance.

To promote the creation of an ever-growing open-source dataset, we have also developed and released a labeling tool[III] to ease the annotation process. The tool leverages the off-the-shelf pose detector OpenPifPaf [95] to locate the 2D bounding boxes of pedestrians. It then runs a pre-trained model (more detailed on Section 6.5) on the JAAD [157] and PIE [154] dataset to provide a first guess. This pipeline allows annotators to only check and eventually correct wrong predictions.

To count the number of pedestrians, we use the tracking identification number for JRDB [121] dataset, while for the KITTI dataset [64] we manually count them. In the case of the nuScenes dataset [31], we leverage the metadata associated with each frame. We approximate the number of unique pedestrians by only counting once the instances that appear in the same camera multiple times in a 5-seconds time window. We run sensibility analysis on the time window and opt for 5 second as the images are recorded from a moving vehicle in the majority of scenes.

Table 6.2 – Comparing our proposed method and baseline results on JAAD [157] and on our LOOK dataset. We evaluate eye contact classification using the average precision (AP) metric. For a fair comparison, we also report the recall of the detected pedestrians for each method. All approaches have been trained for classification on either JAAD solely, or on our LOOK dataset, and we evaluate them on both JAAD and LOOK. Our method is only trained on keypoints and reaches state-of-the-art results on the eye contact detection task on both the JAAD and LOOK datasets when compared with image- and crop-based methods. It also shows the best generalization properties when evaluated on a different dataset. The keypoints are obtained running an off-the-shelf pose estimator [95] without re-training or adapting it to the different datasets.

Dataset	Method	Input	Eye Contact Classification (AP) ↑ [Pedestrian Detection Recall ↑]				
Train / Eval			JAAD	LOOK-KITTI	LOOK-JRDB	LOOK-nuScenes	LOOK
JAAD	Rasouli [157]	Crops	75.4 [80.1]	65.9 [99.8]	87.2 [98.2]	78.7 [89.8]	77.3 [95.9]
	MTL-Fields [126]	Images	82.6 [92.4]	89.7 [93.1]	82.1 [81.9]	92.0 [71.8]	87.9 [82.3]
	Our method	Keypoints	85.9 [80.1]	91.6 [99.8]	94.8 [98.2]	91.0 [89.8]	92.5 [95.9]
LOOK	Rasouli [157]	Crops	71.0 [80.1]	76.8 [99.8]	89.5 [98.2]	82.9 [89.8]	83.1 [95.9]
	MTL-Fields [126]	Images	80.7 [79.0]	95.1 [96.5]	95.2 [93.0]	93.4 [68.4]	94.6 [86.0]
	Our Method	Keypoints	**86.0** [80.1]	**96.4** [99.8]	**97.1** [98.2]	**95.1** [89.8]	**96.2** [95.9]

6.4 Eye Contact Detection

The goal of our method is to detect from images whether humans are looking at the camera or somewhere else. We tackle autonomous driving scenarios, *i.e.*, outdoor scenes where people may be several meters far from the camera. Our approach consists of two steps. First, we use a 2D pose detector to obtain a low-dimensional representation from the image domain, which we call *semantic keypoints*. The keypoints are a convenient representation that provides invariance to many factors, *e.g.*, background artifacts, clothes, weather conditions. Second, we feed the extracted keypoints to a simple feed-forward neural network that detects the presence of eye contact. In addition, we also explore multi-modal representations, by combining the keypoint representations with the features obtained from crops of images, and we explore different fusion techniques. A diagram of our modular architecture can be found in Figure 6.2.

6.4.1 Keypoint-based Method

We escape the image domain using 2D keypoints: a low-dimensional representation obtained through the off-the-shelf pose detector OpenPifPaf described in Chapter 2, which was designed for crowded scenes and low-resolution images. The output of our network is the binary flag indicating whether a person is looking at the camera. To create the training and testing dataset, we match the bounding boxes enclosing the keypoints with the ground-truth bounding boxes using their intersection over union (IoU). We select the instances with the highest matching IoU above 0.3 for each ground truth.

Keypoints are especially useful to prevent overfitting. To further increase generalization properties, we normalize the keypoints and zero-center them on the y-axis. Normalization prevents different scale differences from biasing the results, while the vertical location of a person in the image plane does not add any information regarding eye contact detection. The x-coordinate in the image plane, on the other side, may help infer the relative head and body orientations with respect to the camera. For every keypoint i with pixel coordinates (u_i, v_i) in the image plane, we apply the following transformation:

$$\begin{cases} \hat{u}_i = & \frac{u_i - u_{hip}}{w_{box}} + \frac{u_{hip}}{w_{image}}, \\ \hat{v}_i = & \frac{v_i - v_{hip}}{h_{box}}, \end{cases} \tag{6.1}$$

where $(u_{hip},\, v_{hip})$ is the mean of the coordinates of the left and right hips of the instance, w_{box}, h_{box}, the width and height of the enclosing box given by the keypoints, and w_{image} the width of the input image. In practice, the normalization removes information on the size of the person as well as on the vertical location in the image plane.

Our architecture is composed of a simple fully-connected network with residual blocks [73], and includes batch-normalization [79] after every fully connected layer as well as dropout [183]. The structure is inspired by the success in 3D vision tasks using 2D keypoints, especially Martinez *et al.* [122] for 3D pose estimation, and Bertoni *et al.* [23] for human 3D localization. The residual blocks increase performances and avoid overfitting, while the model, which contains approximately 411K training parameters, is characterized by great speed and a low memory footprint. Its building blocks are shown in Figure 6.2.

6.4.2 Combined Method

We argue that 2D keypoints are a low-dimensional representation that contains enough information to understand whether a person is looking at the camera. This is motivated by the application we are targeting: autonomous driving scenarios, where people are often further away from the camera and the pupils may not be distinguishable. To test our hypothesis, we develop a modular architecture to optionally include visual information from the head region of a pedestrian. We create a combined method that encodes features from both the keypoints and the cropped head region. While the former branch does not change, we select for the crops the upper third of the bounding box [157] enclosing the keypoints, and we extract the features using a convolutional backbone. We explore different backbone architectures (*i.e.*, AlexNet, [96] ResNet [73] and ResNeXt [207]) and different fusion techniques. As visually described in Figure 6.2, we experiment with *early fusion* and with *late fusion*. In the former option (O1), we sum the visual features extracted from a convolutional backbone with the raw features extracted from the 2D keypoints. In the latter option (O2), we concatenate the visual features together with the features extracted from the last layer of the fully-connected architecture. Our training schedule, inspired by Zamir *et al.* [211], consists of two steps. We first initialize the keypoint-based and the crop-based branches by training them independently. We then keep frozen all the layers before the concatenation and train the remaining ones. At both stages, we use the binary cross-

Table 6.3 – Impact of different architectures on the AP metric for eye contact classification (%) on different datasets. We train all the methods on the JAAD dataset [157] only, and we evaluate them on JAAD, PIE [154], and our LOOK dataset. ***Crops*** stands for adapting a crop-based model first introduced by Rasouli *et al*. [157] with a ResNet [73] or ResNeXt [207] architecture. ***Keypoints*** stands for our simple architecture only trained with keypoints as input, either all the 17 keypoints of the human body, or a subset of it: *keypoints - Body* includes all the keypoints but the head ones, while *Keypoints - Head* includes the ears, eyes and nose locations. ***Keypoints & Crops*** stands for our fusion-based approach combining keypoints and crops in a single representation. We experiments with two backbones for crops, ResNet-18 (R-18) [73] and ResNeXt-50 (RX-50) [207], and with two fusion techniques, O1 and O2. When training only using the 5 head keypoints, we obtain the best results on JAAD [157] but training on all the keypoints generalizes better across datasets.

Method	JAAD	PIE	LOOK-KITTI	LOOK-JRDB	LOOK-nuScenes	LOOK
Crops (R-18)	78.1	73.5	76.7	92.0	81.7	83.5
Crops (RX-50)	79.7	74.2	72.0	92.5	85.7	83.4
Eyes Crops	77.4	70.6	77.1	83.6	84.7	81.8
Keypoints	85.9	83.8	**91.6**	94.8	91.0	**92.5**
Body Keypoints	76.4	72.6	79.3	75.4	80.7	78.5
Head Keypoints	**86.3**	**84.0**	90.9	**95.1**	90.2	92.0
Keypoints & Crops (R-18, O1)	78.0	75.2	79.7	91.6	85.3	85.5
Keypoints & Crops (R-18, O2)	78.7	75.6	78.9	92.7	84.3	85.4
Keypoints & Crops (RX-50, O1)	79.5	75.1	73.6	92.1	85.8	83.8
Keypoints & Crops (RX-50, O2)	80.6	75.9	74.1	93.2	86.2	84.5
Keypoints & Eyes Crops (O1)	83.9	79.9	87.0	92.5	**91.2**	90.2

entropy loss. The combined method allows us to verify whether adding visual information to the keypoint-based method increases the performance.

6.5 Experiments

6.5.1 Experimental setup

Evaluation Metrics. We evaluate pedestrian detection and eye contact classification separately. Some previous methods [157] do not include pedestrian detection, using a box classification approach, where the ground-truth boxes are given. In our case, a detection step is also included to ensure fair comparison among different methods. To disentangle the contributions of pedestrian detection and eye contact classification, we split the two tasks. To evaluate the detection results, we use the recall metric with a threshold on intersection over union (IoU) of 0.5. Compared to [54, 126], we do not use Average Precision (AP) metric for detection as we only focus on instances labeled with the eye contact attribute, and in any given dataset very far instances are not annotated for it. In this case, the AP metric may penalize extra detections. In the classification setup, we evaluate the set of detected instances that match a ground-truth, and we use the AP metric to evaluate the classification of the binary attribute of looking or not at the camera.

As shown in Table 6.1, each dataset is unbalanced toward a majority of people not looking at the camera. Following again the procedure of [126], we compute results on a balanced test set where negative instances are randomly sampled. The sampling is done 10 times to reduce the variance and the results are averaged.

Regarding training and testing split, for JAAD dataset [157] the official split is composed of 177 videos for training, 29 videos for validation, and 117 videos for testing. For PIE dataset we use, as recommended, *set01*, *set02*, *set04* for training, *set05 set06* for validation and *set03* for the testing set.

Implementation Details. To obtain input-output pairs of 2D keypoints and binary labels, we apply the off-the-shelf pose detector OpenPifPaf [95, 94] and match our detections with the ground-truth boxes provided by each dataset. We train our keypoint-based architecture for 20 epochs, using binary cross-entropy loss function with Adam optimizer [89], with a learning rate of 0.0001, and mini-batches of 64 instances. The crop model is trained for 20 epochs, using binary cross-entropy loss function, SGD optimizer [29] with Nesterov momentum [133], a learning rate of 0.0001, and mini-batches of 32 instances. For the combined architecture, we pre-train both branches and freeze the early layers before the fusion of the features. We train the last layers with an SGD optimizer, a learning rate of 0.00001, and mini-batches of 128 instances.

The code, available online, is developed using PyTorch [141]. We do not apply any data augmentation procedure on the 2D poses.

6.5.2 Baselines

We argue that eye contact detection is a crucial task yet to be solved to develop safe autonomous vehicles. However, to the best of our knowledge, very few methods have reported results on the eye contact task in JAAD [157] or PIE [154]. Rasouli *et al*. [157] proposed to use image crops of people as inputs to an AlexNet architecture [97] followed by fully connected layers. Their published results on the JAAD dataset [157] used a smaller version of the dataset and randomly split the instances of the dataset. Hence, the same unique pedestrian in different time frames may appear both in training and testing sets. For a fair comparison, we have re-implemented this method and evaluated it on the recently released official JAAD split to prevent any contamination of the testing set.

In addition, we compare against the very recent MTL-Fields developed by Mordan *et al*. [126]. It is a field-based approach that leverages multiple pedestrian attributes in a multi-task fashion, including eye contact, to understand the visual appearances and behaviors of pedestrians. Contrary to Rasouli *et al*. [157] that operate on image crops of people and therefore discard context around them, MTL-Fields keep full images in order to understand the scenes and learn interactions between pedestrians. As their code is open-source, we train a network and evaluate it with our setup for eye contact detection.

Our Baselines. One of our goals is to compare the properties of keypoints and crops for the eye contact task. Thus, we develop a modular architecture that is either based on keypoints only, or can combine keypoints and visual information together, and we benchmark it with the following baselines:

- *Crops*: when referring to methods using crops only, we consider the approach of Rasouli *et al*. [157] with a more recent ResNet [73] or ResNeXt [207] backbone. Rasouli *et al*. [157] train their model with ground-truth crops without including detection results. For a fair comparison, we train and evaluate the model including the same set of instances provided by the OpenPifPaf detector [94].
- *Head & Body Keypoints*: we test our keypoint-based architecture with subsets of keypoints, either only including the keypoints of the head region (eyes, nose, ears), or only the ones from the rest of the body. The goal is to analyze whether the body orientation also provides informative cues, or the head keypoints suffice for the eye contact detection task.
- *Keypoints & Eye Crops*: we test whether adding visual information about the region around the eyes could be informative and less prone to overfitting than head crops. From the 2D keypoint locations of the eyes and ears, we crop a small region around the pupils and resize it to a fixed patch of 3x10x30 pixels. The model architecture is consistent with the one shown in Figure 6.2, but we substitute the head crops with the eyes one, and a convolutional backbone with a fully connected block.

Table 6.4 – Evaluating cross-dataset results for our best crop- and keypoint-based methods using the AP binary classification metric (%) on the JAAD dataset [157]. In parenthesis, the relative difference with respect to the same method trained on the JAAD dataset [157]. *Instances* counts the total number of training instances.

		JAAD [157]	
Training Datasets	Instances	Crops	Keypoints
JAAD [157]	50K	79.7 (-)	85.9 (-)
LOOK-nuScenes [31]	10K	71.1 (-8.6)	84.6 (-1.3)
LOOK	41K	73.7 (-6.0)	86.0 (+0.1)
LOOK + PIE [154]	61K	75.1 (-4.6)	**87.7** (**+1.8**)

6.5.3 Quantitative Results

In Table 6.2, we show the results of training and evaluating each method on the same dataset (either JAAD [157] or our LOOK dataset) as well as cross-dataset results. Our method achieves the best performances when compared to the other baselines, achieving at least a 5% improvement both when testing on the same dataset and when evaluating cross-dataset generalization properties. More notably, our model is able to generalize well on our LOOK dataset when trained on the JAAD dataset [157] only, as it reaches an AP of 92.5%; almost on par when compared against baselines trained on the LOOK dataset. Qualitative examples from different datasets are shown in Figure 6.4.

Our recall results are shared with Rasouli [157] baseline, as we train and evaluate their model on the same instances detected by the off-the-shelf pose detector OpenPifPaf [95]. We observe that MTL-Fields [126] achieves higher recall on JAAD when trained on the same dataset, but the same recall drops by 15% when trained on a different dataset. Our method on the other side maintains high recall when testing domain adaption, as it leverages an off-the-shelf pose estimator [95] trained and optimized on a general-purpose dataset [111] that is beneficial for domain adaptation.

6.5.4 Cross-dataset Generalization.

We further study cross-dataset generalization with our new LOOK dataset in Tables 6.3 and 6.4. First of all, we investigate the performances of eleven crop- and keypoint-based methods in Table 6.3. We train the models on JAAD dataset [157] and evaluate them on JAAD [157], PIE [154], and our LOOK dataset. Keypoint-based models perform and generalize better than crop-based ones, consistently with results obtained in Table 6.2. Surprisingly, combining visual information to the keypoints into our combined models (which we refer to as *Keypoints & Crops*) degrades the performances as it leads to stronger overfitting. The best results are achieved only by combining features from the eye region instead of the head region. We attribute this result to the generalization properties of keypoints, as this low-dimensional representation does not overfit

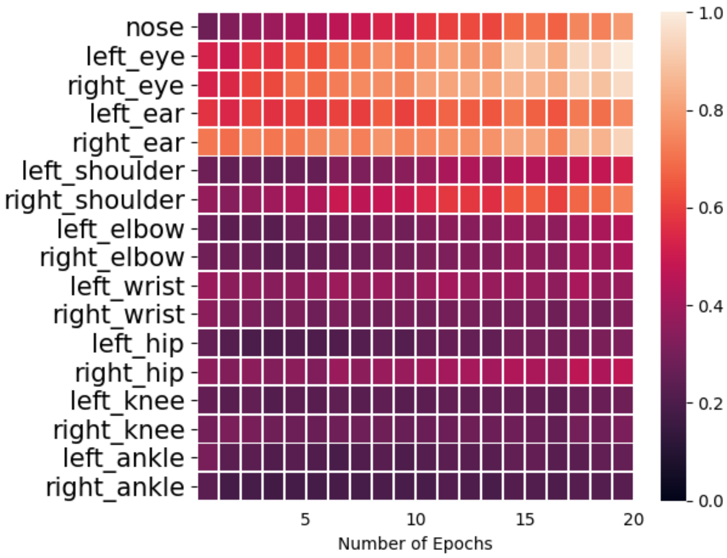


Figure 6.3 – Visual illustration of the normalized magnitude of the gradients of the loss function with respect to each keypoint during training. The keypoints related to the head (eyes and ears) are the ones that most affect the loss function.

on background scenes or specific faces. Yet even simpler models with no visual information achieve the best results on all the datasets, but for LOOK-nuScenes [31].

As additional experiment in Table 6.4, we train our best keypoint-based and crop-based methods on different datasets but JAAD [157], and evaluate them on JAAD. We train our keypoint-based model on 10K instances from the nuScenes dataset [31], and we obtain less than 2% difference compared to training it on the 50K instances of the JAAD dataset [157]. The best crop-based model, on the other side, achieves an AP of 71.1%, down from an original 79.7%. When increasing the number of instances from multiple datasets, the performances of the crop-based model never reach the baseline result of being trained on JAAd only. The keypoint-based model, on the other side, achieves the best performances on JAAD [157] when trained on different datasets.

6.5.5 Additional Studies

The Role of Distance. We test the hypothesis that crop-based methods may be most effective when people are closer to the camera, while keypoints may be more useful when people are far away and details of the face are less informative. We obtain the distribution of bounding box heights for all the instances of the JAAD test set [157], and evaluate each quartile separately. As we show in Table 6.5, the hypothesis is not verified: keypoints remain more effective than crops even for people close to the camera. This result may not be intuitive at first sight, but we are analyzing autonomous driving datasets, where even close people may be several meters away

Table 6.5 – Average precision (AP) in percentage (%) as a function of the bounding box height in pixels for the JAAD dataset [157]. Each cluster corresponds to one quartile of the distribution. For the crop-based methods, we consider our ResNeXt-50 model with late fusion when not differently specified.

	JAAD [157]				
Method / Box Height [px]	240+	160-240	110-160	0-110	All
Crops (ResNet-18)	80.4	79.5	80.4	73.4	78.1
Crops (ResNeXt-50)	79.0	81.2	83.0	75.3	79.7
Eyes Crops	74.4	79.1	81.3	74.7	77.4
Keypoints	87.9	88.7	**87.0**	78.7	85.9
Head Keypoints	**90.4**	**89.7**	86.7	76.5	**86.3**
Keypoints & Crops	80.5	82.2	83.6	75.9	80.6
Keypoints & Eyes Crops	85.3	86.4	85.0	**79.4**	83.9

from the camera, and detecting the direction of the pupil may not be feasible. Keypoints provide a simple yet effective representation in these scenarios.

Saliency Map. To verify the impact of each keypoint in the final decision of the model, we compute the absolute value of the gradient of the objective function with respect to each input node for every epoch on the training set:

$$i_k = \frac{1}{N}\sum_{j=0}^{N}\left|\frac{\partial L(y_j, f(x_j))}{\partial k_x}\right| + \left|\frac{\partial L(y_j, f(x_j))}{\partial k_y}\right| + \left|\frac{\partial L(y_j, f(x_j))}{\partial k_c}\right|, \tag{6.2}$$

where i_k represents the impact of the keypoint k with its three components: k_x and k_y coordinates, and the confidence score k_c. We then average this value by taking the mean absolute value across all the training instances (x_j, y_j) that consists of N samples. The results are illustrated in Figure 6.3. The dominant keypoints are the eyes and ears, as shown by the magnitude of the gradients of the loss function L with respect to each keypoint.

Run Time. Our experiments have been conducted using a machine with a single NVIDIA GeForce GTX 1080 Ti and Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz. In Table 6.6, we compare the run time performances of several methods on the test set of the JAAD dataset [157]. As the original method from Rasouli *et al*. did not include a detection step, we use the same backbone to extract the poses for our method and the crops for Rasouli's one. For MTL-Fields, detection and classification are performed in a single stage. Our method excels in the classification step with less than 1 ms of inference time as it uses low-dimensional keypoints. Regarding the detection step, our method is agnostic to the pose detector. We have tested it with OpenPifPaf

Table 6.6 – Average run time performances for a single image on the JAAD test set. The detection steps for both Rasouli [157] and our method are calculated with the off-the-shelf pose detector OpenPifPaf [95], using either a ResNet-50 (R-50) [73] or a ShuffleNetV2K30 (S-30) [118] backbone.

		Average Run Time (ms)		
Method	AP (%)	Detection	Classification	Total
Rasouli [157] (S-30)	71.0	602	39	672
MTL-Fields [126]	80.7	–	–	573
Our Method (S-30)	**85.9**	602	0.8	626
Our Method (R-50)	82.6	**305**	0.8	**328**

[95] using two different backbones and achieved the fastest run time with a ResNet-50 (R-50) [73].

(a) LOOK-KITTI [64]

(b) LOOK-nuScenes [31]

(c) JAAD [157]

(d) LOOK-JRDB [121]

Figure 6.4 – Qualitative results for the eye contact detection task on multiple datasets. People with green poses are predicted as looking at the camera, people with red poses as not looking.

6.6 Beyond Eye Contact: Simple Yet Effective Action Recognition for Autonomous Driving

This chapter shows that our keypoint-based method achieves state-of-the-art results on detecting eye contact, a communicative action where one could expect images to be a crucial input representation. In this section, we expand our focus beyond this communicative action and shed light on the effectiveness of semantic keypoints for action recognition in the context of autonomous driving. First, we validate the design of our model on the TCG Dataset [201] that collects accurate 3D body keypoints for recognition of traffic control gestures. Using ground-truth keypoints as input lets up validate our architecture filtering out the impact of keypoints' quality. Second, we focus on cyclists by assessing arm signal recognition on the CASR dataset [56]. Finally, we analyze the complete action recognition workflow starting from raw images with both pedestrians and cyclists. We train and evaluate our approach on various action recognition tasks on TITAN [119], a new dataset captured from a moving vehicle on highly interactive urban traffic scenes in Tokyo.

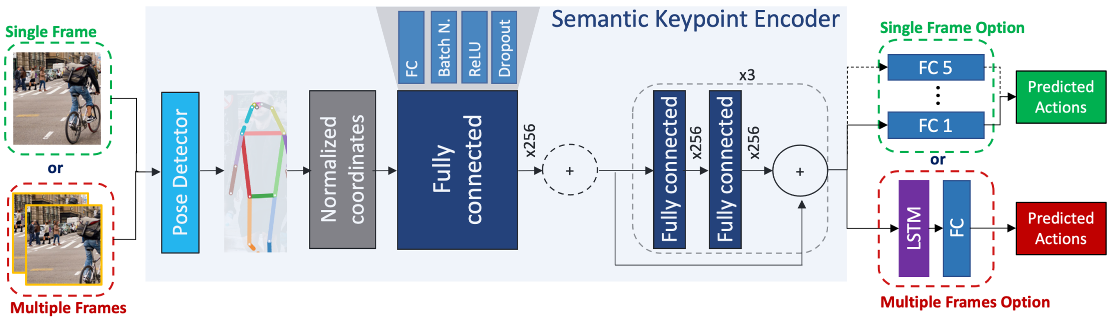


Figure 6.5 – Model architecture for action recognition tasks. The input is a set of 2D keypoints extracted from a raw image and the output is the estimated action of a pedestrian. We use three different heads for image-based and video-based action recognition, and for estimating simultaneous actions (*e.g.*, a person may be walking while talking at the phone).

We use the same architecture described in Section 6.4, encoding the keypoints with a feedforward network, and predicting the corresponding action from the encoded representation. We also extend our method (i) to estimate actions from videos and (ii) to predict simultaneous groups of actions (*e.g.*, a person can walk while being on the phone). For (i), we add a simple LSTM [74] to process a temporal sequence of poses before the final linear layer, and for (ii), we use a multitask approach, where multiple parallel heads (instead of just one) process a shared representation to yield multiple predictions. We train our architecture with focal loss [110]. The architecture is illustrated in Figure 6.5.

6.6.1 Experiments

Traffic Control Gestures. We evaluate traffic control gestures on the TCG dataset [201], which consists of 250 sequences from five actors staging to regulate the traffic on road intersections. The

Table 6.7 – Action recognition results on TCG [201] test set. Our single-frame model performs on par with all the simple temporal baselines while being outperformed by a complex attention-enhanced Adaptive graph convolutional network [143].

Method	Cross-subject			Cross-view		
	Accuracy (%)	Jaccard (%)	F1 (%)	Accuracy (%)	Jaccard (%)	F1 (%)
RNN	82.81	57.40	69.45	80.94	57.21	69.98
GRU	84.44	58.16	70.45	83.47	56.25	68.59
LSTM	83.23	56.32	68.59	79.58	52.02	64.62
Att-LSTM	85.67	50.70	61.87	85.30	59.87	71.20
Bi-GRU	86.80	57.25	68.95	87.37	55.55	67.68
Bi-LSTM	87.24	67.00	78.48	86.66	65.95	77.14
TCN	83.44	62.06	74.23	82.66	63.97	75.95
GCN	65.42	38.55	50.73	62.40	35.05	48.51
AAGCN [175]	**91.13**	-	85.81	90.22	-	85.21
AAGCN++ [143]	91.09	-	**86.26**	**90.64**	-	**85.52**
Ours (single-frame)	85.03	63.72	76.91	86.29	68.76	80.81
Ours (temporal)	87.31	**69.15**	81.15	87.74	**70.11**	81.89

actors use the standard European traffic control gestures, *i.e.*, stop, go and clear. Following the cross-subject and cross-view evaluation protocols in TCG, Table 6.7 compares the performances of our single-frame and temporal models with eight simple baseline methods as well as two more complex attention-based graph convolutional networks [175, 143]. Our temporal model outperforms the simple baseline models. Specifically, it performs better than the LSTM baseline, which directly predicts actions from raw keypoint coordinates. This demonstrates the effectiveness of processing raw keypoints with a feedforward network. Our models are still outperformed by the two attention-based graph convolutional networks, but those are much heavier and would likely be less suitable for applications with hard run-time constraints (e.g., memory footprint, inference time), such as autonomous driving. Additionally, traffic control gestures are designed to be unambiguous actions that could easily be understood without temporal context, which means temporal information should not be crucial for these gestures. The close results of our single-frame and temporal models confirm this observation.

Table 6.8 – Assessing the performances of our method on the CASR dataset [56]. We evaluate cyclists' arm signals, *i.e.*, turning left, right, or stopping using classification accuracy and F1 score.

Method	Accuracy (%) ↑	F1 Score (%) ↑
Fang et al. [56]	93.0	**92.0**
Ours (single-frame)	**93.4**	91.2

Cyclist Arm Signal Recognition. Cyclists, like pedestrians, are vulnerable road users prone to injury in any vehicular collision. They usually communicate their intentions with arm signals, indicating whether they are turning left, right, or stopping. Hence, we evaluate the performances of our method to assess arm signal recognition. We use the CASR dataset [56], which contains annotated videos of 219 cyclists' actions. Each frame only contains a single cyclist in front of the camera, simplifying the evaluation procedure. Our method is similar to the baseline of Fang et al. [56] as they also use a pose detector to extract 2D keypoints. Out of each keypoint set, they hand-craft a set of 1170 features based on angles and distances between keypoints and use a Random Forest classifier to process them. In our case, we simply pass the normalized keypoints to a light-weight neural network. Results are shown in Table; our method achieves comparable results with Fang et al. [56], both reaching an accuracy of 93%.

Table 6.9 – Action recognition results on TITAN [119] test set. Our single-frame method outperforms a 3D ResNet [69] using the accuracy suggested in the original dataset evaluation [119]. In reality, every method's predictions are highly imbalanced towards the *no action* class, as the average precision metric suggests.

	I3D [36]	3D ResNet [69]	Ours (multitask)	
Action group	Accuracy (%)	Accuracy (%)	Accuracy (%)	mAP (%)
atomic	92.19	75.52	80.01	26.80
simple	53.18	31.73	47.97	20.27
complex	98.81	98.80	97.80	15.50
communicative	86.49	86.48	83.69	29.55
transportive	90.80	90.81	89.80	28.30
overall	84.29	76.67	79.85	24.08

Table 6.10 – Action recognition results with selected actions on TITAN [119] test set. We focus on atomic actions that can be perceived using semantic keypoints. Our method strongly outperforms the crop-based baseline for every task, especially on the most challenging actions, such as *bending* and *sitting*.

		Average Precision (AP %) ↑ [Detection Recall ↑]					
Method	Inputs	Walking [75.4%]	Standing [65.4%]	Sitting [62.1%]	Bending [71.8%]	Biking [82.4%]	Average [73.7%]
Ours (multitask)	Keypoints	90.16	40.67	43.78	41.12	57.70	48.14
ResNet-50 (He et al.)	Crops	92.85	42.07	5.18	8.44	56.00	40.91
Ours (single-frame)	Keypoints	96.87	64.55	**81.22**	**64.59**	**88.30**	**79.11**
Ours (temporal)	Keypoints	**97.83**	**73.02**	65.78	47.31	84.98	73.78

Action Recognition for Autonomous Driving. After validating our method on traffic control gestures and arm signals, we focus on general action recognition for autonomous driving applications. The TITAN dataset [119] has 700 video clips captured by an on-board camera that include both pedestrians and cyclists, making it a suitable dataset for evaluating the complete recognition workflow starting from raw images. All annotated actions belong to *atomic*, *simple context*, *complex context*, *communicative* or *transportive* action groups. Notably, all the people in all the frames are annotated with five action labels, i.e., one from each action group (including labels for *no action* for any group).

In Table 6.9, we compare our multitask model, using five prediction heads to match the five action groups of TITAN, with I3D [36] and 3D ResNet [69]. We observe it has comparable accuracy to the other two methods. However, TITAN is highly imbalanced (toward *no action* for almost all groups), thus the overall accuracy is not a suitable metric to evaluate results. For example, in the complex context action group, more than 98% of the persons are labeled as *no action*, and each method overfits on this class. For this reason, we propose to evaluate using mean Average Precision (mAP), where Average Precisions (APs) are computed for all classes separately and then averaged, as seen in Table 6.9, while our method reaches an accuracy of 80%, outperforming a 3D ResNet[69], it only achieves a mAP of 24%, confirming our hypothesis that all the predictions are highly imbalanced towards the *no action* class.

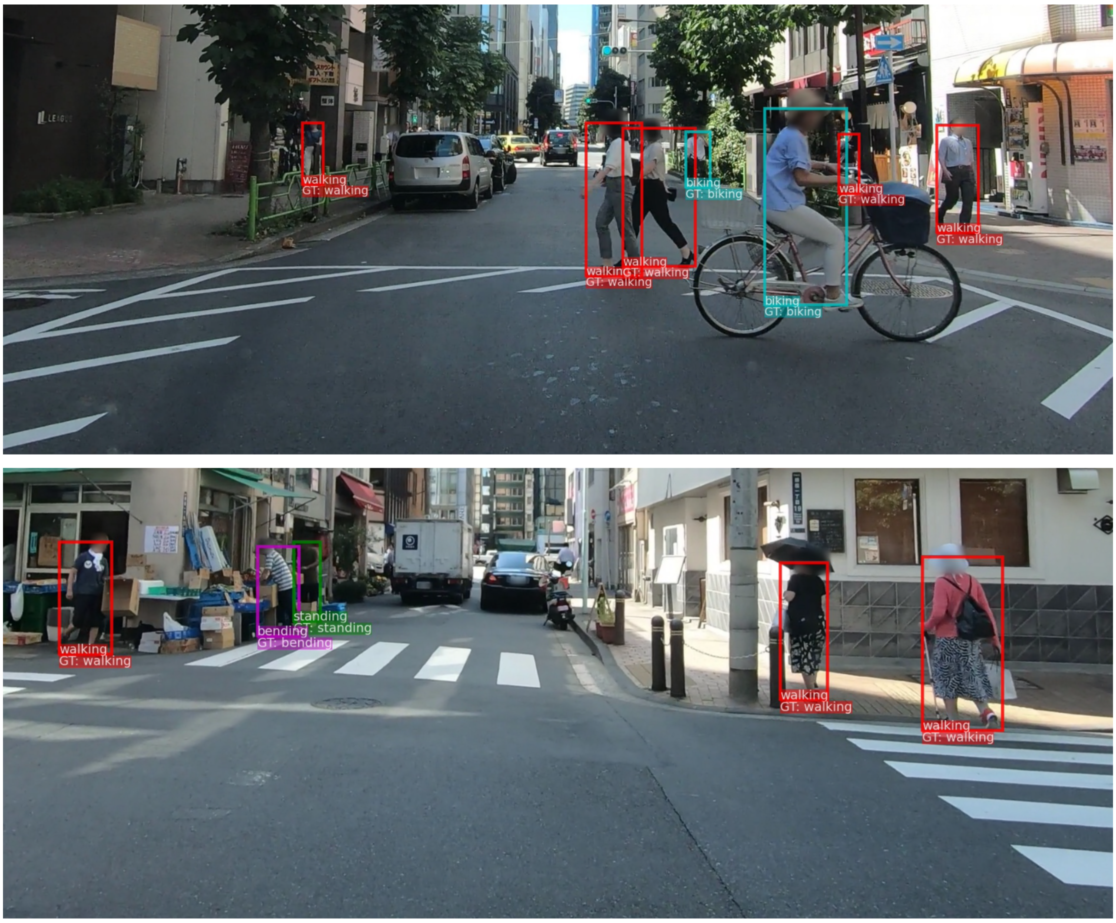


Figure 6.6 – Action recognition examples from our single-frame model on TITAN [119] test set. Predicted actions and ground truths (GT) are shown at the bottom of the boxes. Each color represents a different action, *e.g.*, red for *walking* and purple for *bending*.

Since the labels *no action* dominate most action groups, and some actions have insufficient numbers of examples, we focus on a subset of actions where our multitask model has reasonable

Figure 6.7 – Examples of failure from our single-frame model on TITAN [119] test set. Left image: without temporal context, discriminating between *walking* and *standing* is hard, especially with occlusions. Middle image: contextual features would help distinguish that the person is not biking even if the pose resembles it. Right image: the man is picking up an object, an indication of *bending* and not just *standing*.

mAP. We also merge actions *cycling* and *motorcycling* with close meaning, resulting in five classes of interest: *walking*, *standing*, *sitting*, *bending* and *biking*. Table 6.10 compares the recognition performances of four models using mAP. Our multitask model is trained on the original TITAN dataset, and we only keep the predictions corresponding to the selected action subset. Since the original dataset contains considerable *no action* samples, the training process of this model is dominated by this majority class, and the model does not have satisfactory mAP. The following three models are trained and tested on the selected action subset. The first of them is a ResNet-50 [73] classification network trained on image crops centered on detected pedestrians. For our temporal model, we follow the procedure used in TITAN [119] and obtain temporal sequences by associating detected poses using ground-truth track IDs. The results confirms the findings of Table 6.2 for the eye contact detection task: our keypoint-based model strongly outperforms the crop-based one. This is particularly evident for *sitting* and *bending* classes. The dataset contains only 304 and 1297 instances for these classes, out of a total of 50K instances, and the low-dimensionality of semantic keypoints helps in low-data regimes. The results also highlight that our temporal model is better at *walking* and *standing*, two visually close actions where temporal context should help disambiguation, while our single-frame model is better for *sitting*, *bending* and *biking*, for which temporal information is not as important. Figure 6.6 presents qualitative examples from our single-frame model, and Figure 6.7 illustrates failure cases, where combining image-based features and pose estimation could help improve the results.

6.7 Conclusion

This chapter has introduced a deep learning approach for human-robot eye contact detection in the wild, *i.e.*, with no prior knowledge of the environment, suited to the multiple challenges associated with this task. We have discussed the importance of eye contact to forecast human behaviors and publicly released LOOK, a large-scale dataset for human-robot eye contact detection in the wild. We designed it with real-world generalization in mind by annotating three standard autonomous driving datasets to consider cross-dataset training and evaluation and focus on multiple scenarios and diverse environments. We evaluated our method and several approaches from the literature on LOOK to create a benchmark for this task and show state-of-the-art results with robust generalization across datasets compared to image-based approaches. Finally, we have demonstrated that our method generalizes to different action recognition tasks, moving towards a holistic framework for action recognition in the wild.

The main limitation of our approach is that instances are analyzed independently. However, some actions are performed in groups and require modeling people's interactions. *Talking* is the most evident example: two or more people close to each other may just be standing waiting for the traffic light signal, or they may be talking with each other and not be interested in crossing the street. In the next chapter, we will combine our studies on 3D localization and action recognition to study social interactions among humans. We will focus on the field of proxemics, analyzing how people arrange themselves in the space [87] and inferring social relations. In addition, we will demonstrate the applicability of our approach beyond autonomous driving applications. We will show how our neural network architecture can be used to fight the COVID-19 outbreak by monitoring social distancing in a privacy-safe manner.

7 Beyond Autonomous Driving: Social Interactions and Social Distancing

This chapter is based on the article:

Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi, *Perceiving Humans: From Monocular 3D Localization to Social Distancing*, IEEE Transactions on Intelligent Transportation Systems, 2021

Additional qualitative results are displayed on YouTube[I].

This story appeared in the following media:

- BBC 4Tech show[II]
- EPFL News [III]
- SwissInfo[IV]
- LeMatin[V]

7.1 Introduction

The previous chapters study the effectiveness of semantic keypoints to localize humans in 3D and to detect their actions for autonomous driving applications. One limitation of our approaches is that every instance is analyzed independently, while certain types of action, such as *talking*, are performed in group. This chapter combines our 3D localization and action recognition findings to propose a deep learning approach that perceives humans and their social interactions

[I] https://www.youtube.com/watch?v=r32UxHFAJ2M
[II] https://www.youtube.com/watch?v=0BkQEOwa5kc&list=PL63lwGZ_8vsnmWuxFBTL6NulPaYZXj0zv&index=1&t=61s
[III] https://news.epfl.ch/news/3d-detectors-measure-social-distancing-to-help-fig/
[IV] https://www.swissinfo.ch/eng/3d-detectors-measure-social-distancing-to-help-fight-covid-19-/46601518
[V] https://www.lematin.ch/story/lepfl-a-une-solution-pour-nous-faire-garder-nos-distances-523241072624

Figure 7.1 – Our method retrieves 3D locations with confidence intervals, body orientations, social interactions and social distancing in the wild from a single RGB image. We leverage 2D semantic keypoints as intermediate representations, which allow to verify social distancing compliance while preserving privacy.

in the 3D space from visual cues only. Perceiving when and how people interact is essential to understanding mobility patterns and forecasting human behavior in transportation applications. At the same time, we prove our method to be valuable beyond autonomous driving applications by measuring social distancing to help fight COVID-19.

The COVID-19 pandemic has forced authorities to limit non-essential movements of people, especially in crowded areas or public transports [16]. Social distancing measures are becoming essential to restart passenger services, *e.g.*, leaving train seats unoccupied. Yet, in many contexts, it is not obvious how to preserve inter-personal distances. When the risk of contagion remains, we should work to minimize it, and perceiving social interactions can play a vital role in this quest. In fact, talking with a person does not incur the same risk of infection than crossing someone in the street. The infection rate of a disease can be summarized as the product of exposure time and exposure to virus particles [161, 30]. When people are talking together, not only the exposure time escalates, but the act of speaking itself increases the release of respiratory droplets about 10 fold [14, 184]. These analyses urge us to rethink safety measures and focus on proximal social interactions, which can be defined as any behavior of two or more people mutually oriented towards each other's that influence or that take account of each other's subjective experiences or intentions [167]. We show that we can monitor the concept of "social distancing" as a form of social interaction in contrast to a simple location-based rule or smartphone-based solutions [216, 212, 83]. Few methods have studied interactions from images [45, 46], but their results are either limited to personal photos, [209], indoor scenarios, [3], or necessitate a homography calibration [45, 46]. However, the study of social distancing requires an understanding of social interactions in a variety of unconstrained scenarios, either outdoor or within large facilities.

Our approach builds on top of the architectures described in previous chapters and consists of three main steps. First, we use the off-the-shelf pose detector proposed in Chapter 2 to obtain semantic keypoints of humans. Second, the 2D keypoints are fed into our MonoLoco++ architecture (proposed in Chapter 3), which predicts 3D locations, orientations, and corresponding confidence intervals for each person. Finally, driven by these perception tasks, we aim at investigating how people use the space when interacting in groups. According to the subfield of proxemics, people tend to arrange themselves spontaneously in specific configurations called F-formations [87]. The detection of F-formations is critical to infer social relations [45, 46]. Our intuition is that knowing the 3D location and orientation of people in a scene allows retrieving F-formations accurately with simple probabilistic rules. Inspired by [45, 46], we exploit our predicted confidence intervals to develop a simple probabilistic approach to detect F-formations and social interactions among humans. Consequently, we show that we can redefine the concept of social distancing to go beyond a simple measure of distance. We provide simple rules to verify safety compliance in indoor/outdoor scenarios based on the interactions among people rather than their relative position alone. Finally, the design of our pipeline encourages privacy-safe implementations by decoupling the image processing step. Our network is trained on and performs inference with anonymous 2D human poses. An example is provided in Figure 7.1, where 3D location, orientation, and interactions among people are analyzed to verify social distancing compliance in a private manner. The code[VI] and a video[VII] showing qualitative results are available online.

7.2 Related Work

In this chapter, we tackle the high-level task of understanding 3D spatial relations among humans from a single RGB image without ground plane estimation. While perception tasks have been monopolized by relatively new deep learning algorithms, the study of social interactions is based on historic discoveries in behavioural science [81]. In this section, we focus on the subfield called *proxemics*, which investigates how people use and organize the space they share with others [67, 46]. People tend to arrange themselves spontaneously in specific configurations called F-formations [87]. These formations are characterized by an internal empty zone (o-space) surrounded by a concentric ring where people are located (p-space). According to Kendon [87]: "*an F-formation arises whenever two or more people sustain a spatial and orientational relationship in which the space between them is one to which they have equal, direct, and exclusive access*".

These formations characterize how people use the space when interacting to each other. They are characterized by three types of social spaces [67, 45]:

1. *o-space*: An circular empty region to preserve the intimacy of people around it. Every participant looks inward and no people are allowed inside. The type of relation (*e.g.*, personal or business-related) defines the dimensions of this space

2. *p-space*: a concentric ring around the o-space that contains all the participants

3. *r-space*: the area outside the p-space

[VI] https://github.com/vita-epfl/monoloco
[VII] https://www.youtube.com/watch?v=r32UxHFAJ2M

In the case of two participants, typical F-formation are vis-a-vis, L-shape, and side-by-side. For larger groups, a circular formation is typically formed [102]. An example of F-formation configuration is shown in Figure 7.3.

To the best of our knowledge, Cristani *et al*. 2011a [45] is the first work to focus solely on visual cues to discover F-formations and social interactions. In parallel, Cristani *et al*. 2011b [46] studied how people get closer to each other when the social relation is more intimate. Following works have proposed various techniques to automatically detect F-formations in heterogeneous real crowded scenarios [189, 20, 192, 174]. In all approaches is evident how the detection of F-formations is critical to inference social relations and we decided to follow their lead. This line of work, however, considers as input the position of people on the ground floor and their orientation [46] or requires an homogrpaphy estimation to compute the x-y-z coordinates of pedestrians [45]. On the contrary, our approach works end-to-end from a single RGB image. The perception stage, *i.e*., extracting 3D detections from a monocular image, is arguably the most challenging one due to the intrinsic ambiguity of perspective projections.

Social interactions have been also studied in the context of personal photos [209] or egocentic photo-streams [2, 3]. Both approaches assumes humans to stand less than few meters apart from each other and the camera, and do not scale to long range applications, such as monitoring an airport terminal. Recently, deep learning approach has been adopted to understand social interactions under a different perspective. Joo *et al*. [81] learned to predict behavioral cues of a target person (*e.g*., body orientation) from the position and orientation of another person. They aimed to learn the dynamics between social interactions in a data-driven manner, laying the foundations for deep learning to be applied in the field of behavioral science.

7.3 Method

The goal of our method is to recognize social interactions and monitor social distancing given a single image as input. Figure 7.2 illustrates our overall method, which consists of three main steps. First, we exploit a pose detector to escape the image domain and reduce the input dimensionality. Second, we use the 2D joints as input to a feed-forward neural network which predicts x-y-z coordinates and the associated uncertainty, orientation, and dimensions of each pedestrian. Third, the network estimates are combined to obtain F-formations [67] and recognize social interactions.

7.3.1 Social Interactions and Distancing

We identify social interactions by recognizing the spatial structures that define F-formations (see Section 7.2 for more details). Our approach considers groups of two people in an “all-vs-all” fashion by studying all the possible pairs of people in an image.

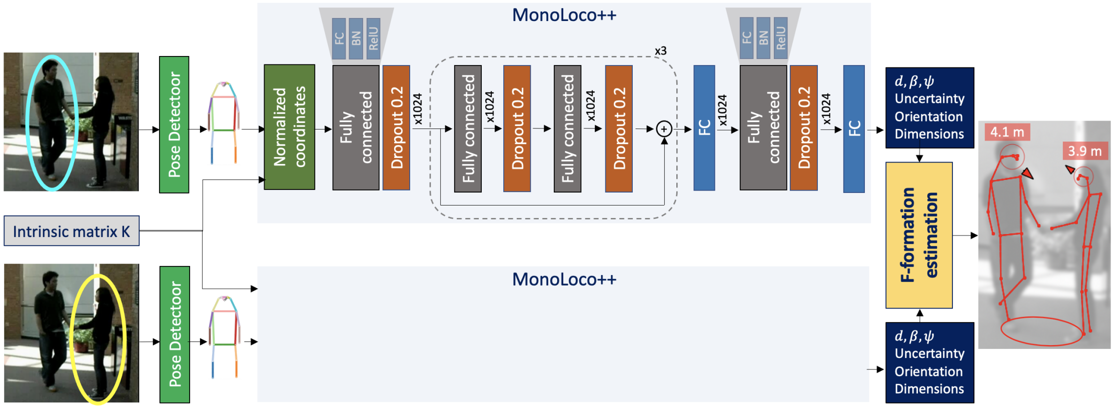


Figure 7.2 – Network architecture. **MonoLoco++** [24]: The input is a set of 2D joints extracted from a raw image and the output is the 3D location, orientation and dimensions of a pedestrian and the localization uncertainty. 3D location is estimated with spherical coordinates: d, azimuthal angle β, and polar angle ψ. Every fully connected layer (FC) outputs 1024 features and is followed by a Batch Normalization layer (BN) [79] and a ReLU activation function. **Social interactions/distancing**: estimates from MonoLoco++ are analyzed with an *all-vs-all* approach to discover F-formations using Eq. 7.2.

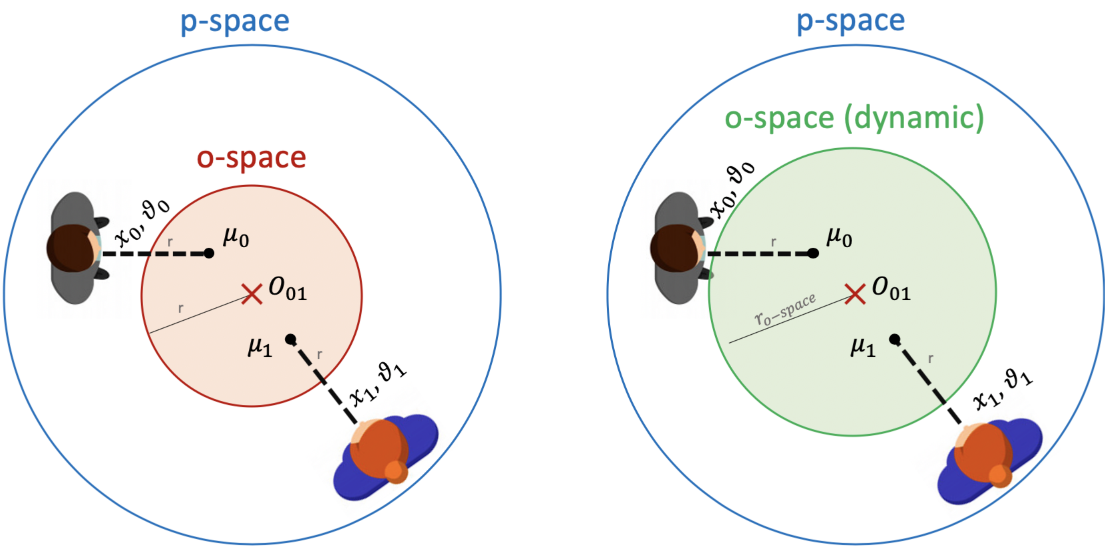


Figure 7.3 – Illustration of the o-space discovery using [45] on the left and our approach on the right. Both approaches use the candidate radius r to find the center of the o-space, as infinite number of circles could be drawn from two points. Differently from [45], once a center is found, we dynamically adapt the final radius of the o-space $r_{o-space}$ depending on the effective location of the two people.

Social Interactions. Ideally, two people talking to each other define the same o-space by looking at its center. In practice, 3D localization and orientation of people are noisy and previous methods [45, 46] have adopted a voting approach. They define a candidate radius r of the o-space and each person vote for a center. The average result defines the center of the o-space. In Cristani *et al*. [45], the candidate radius r remains the final radius of the o-space and is fixed for every group of people. However, once the o-space center is found, nothing prevents us from considering its radius $r_{o-space}$ dynamically as the minimum distance between the center and one of the two people. An illustration of the differences is show in Figure 7.3. Therefore, given the location of two people in the x-z plane $\mathbf{x}$ and their body orientation θ, we define the center and the radius of the o-space as:

$$\begin{aligned} \mathbf{O_{01}} &= \frac{\mu_0 + \mu_1}{2} \\ r_{o-space} &= min(|\mathbf{O_{01}} - \mathbf{x_o}|, |\mathbf{O_{01}} - \mathbf{x_1}|) \quad , \end{aligned} \tag{7.1}$$

where $\mathbf{O_{01}}$ and $r_{o-space}$ are the center and radius of the resulting o-space, μ_0 and μ_1 indicate the location of the two candidate centers of the o-space. In general, $\mu = [x + r * cos(\theta), z + r * sin(\theta)]$ and is parametrized by the candidate radius r, which depends on the type of relation (intimate, personal, business, etc.) [67].

Once the o-space is drawn, we verify the following conditions:

$$\begin{aligned} &(a) \;\; |\mathbf{x_o} - \mathbf{x_1}| < D_{max} \\ &(b) \;\; |\mathbf{O_{01}} - \mathbf{x_i}| < r_{o-space} \;\; \forall i \neq 0, 1 \\ &(c) \;\; |\mu_0 - \mu_1| < R_{max} \end{aligned} \tag{7.2}$$

where D_{max} and R_{max} are the maximum distances between two people, and between the candidate centers of the o-spaces, respectively. Vectors are represented in bold.

The above conditions verify the presence of an F-formation, as:

(a) examines whether two people stand closer than a maximum distance D_{max}, *i.e.*, they lie inside a p-space,

(b) examines whether the o-space is empty (no-intrusion condition),

(c) examines whether the two people are looking inward the o-space.

We note that condition (c) is empirical as looking inward is a generic requirement. Two people usually look at each other when talking, but the needs for social distancing may be different. Our goal is not to find perfect empirical parameters for F-formations discovery, but rather to show the effectiveness of combining simple rules and estimating 3D localization and orientation.

We consider two people as interacting with each other if the three conditions are verified. This method is automatically extended to larger groups as two people can already cover any possible F-formation (vis-a-vis, L-shape and side-by-side), while three or more people usually form a circle [45]. Further, we are not interested in defining the components of each group but rather in understanding whether people are interacting or not.

Social Distancing. The procedure to monitor social distancing can either follow the same steps, or can be adapted to a different context. Risk of contagion strongly increases if people are involved in a conversation [184, 30]. Therefore, recognizing social interactions lets the system only warn the people that incur in the highest risk of contagion. In crowded scenes, this is crucial to prevent an extremely high number of false alarms that could undermine any benefit of the technology. Yet social distancing conditions can also be differentiated from the social interaction ones. For example, a third person invading the o-space could mean that the three people involved are not conversing, but still they may be at risk of contagion due to the proximity. How strict these rules should be can only be decided case by case by the competent authority. Our goal is to help assessing the risk of contagion not only thorough distance estimation but also leveraging social cues.

Uncertainty Estimation. A deterministic approach can be very sensitive to small error in 3D localization, which we know are inevitable due to the perspective projection. Therefore, we introduce a probabilistic approach that leverage our estimated uncertainty to increase robustness towards 3D localization noise. We note that Cristani *et al.* [45] also adopted a probabilistic approach injecting uncertainty in a Hough-voting procedure. However, the chosen parameters were driven by sociological and empirical considerations. In our case, uncertainty estimates comes directly as an output of the neural network and they are unique for each person. Recalling from Section 3.4 that the location of each person is defined as a Laplace distribution parametrized by d and b in Eq. 3.4, we draw k samples from the distribution. For each pair of samples, we verify the above conditions for social interactions. Combining all the results, we evaluate the final probability for a social interaction to occur.

7.4 Experiments

7.4.1 Social Interactions

To evaluate social interactions we focus on the activity of *talking*, which is considered as the most common form of social interactions [45]. From single images, we can evaluate how well we recognize whether people are talking or just passing by, walking away etc.

Table 7.1 – Accuracy in recognizing the *talking* activity on the Collective Activity dataset [43]. In all cases the distance has been estimated by our MonoLoco++. "W/o Orientation", does not uses the estimated orientation, while "Deterministic" leverages orientation but not the uncertainty. "Task Error Uncertainty" refers to the distance-based uncertainty due to ambiguity in the task (Eq. 3.2), "MonoLoco++ Uncertainty" refers to the instance-based uncertainty estimated by our MonoLoco++.

Method	Accuracy (%) ↑	Recall (%)↑
W/o Orientation	67.0	97.2
Deterministic	83.7	97.2
Task Error Uncertainty	91.3	97.2
MonoLoco++ Uncertainty	**91.5**	97.2

Datasets. We evaluate social interactions on the Collective Activity Dataset [43], which contains 44 video sequences of 5 different collective activities: *crossing*, *walking*, *waiting*, *talking*, and *queuing* and focus on the *talking* activity. The *talkin*g activity is recorded for both indoor and outdoor scenes, allowing to test our 3D localization performances in different scenarios. Compared to other deep learning methods [32, 15, 63], we analyze each frame independently with no temporal information, and we do not perform any training for this task, using all the dataset for testing.

Evaluation. For each person in the image, we estimate their 3D localization confidence interval and orientation. For every pair of people, we apply Eq. 7.1 and Eq. 7.2 to discover the F-formation and assess its suitability. We use the following parameters in meters: D_{max}= 2 as maximum distance, and $r_1 = 0.3$, $r_2 = 0.5$ $r_3 = 1$ as radii for o-space candidates. These choices reflect the average distances of *intimate relations*, *casual/personal relations* and *social/consultive* relations, respectively [67].

How much people should look inward the o-space (to assume they are talking) is also an empirical evaluation. We set the maximum distance between two candidate centers $R_{max} = r_{o-space}$ for simplicity. We treat the problem as a binary classification task and evaluate the the detection recall and the accuracy in estimating whether the detected people are talking to each other. To disentangle the role of the 2D detection task, we report accuracy on the instances that match a ground truth. To avoid class imbalance, we only analyzes sequences that contain at least a person talking in one of its frames. Consequently, we evaluate a total of 4328 instances, of which 52.8 % is talking.

Voting Procedure. To account for noise in 3D localization, we sample our results from the estimated Laplace distribution parametrized by distance d and spread b (Eq. 3.4). Each sample vote for a candidate center μ and we accumulate the voting. If an agreement is reached within at least 25% of the samples, we consider the target pair of people as involved in a social interaction and/or at risk of contagion. MonoLoco++ estimates a unique spread b for each pedestrian,

Table 7.2 – Accuracy in monitoring social distancing on KITTI dataset [64]. In all cases the distance has been estimated by our MonoLoco++. "W/o Orientation", does not uses orientation to account for social distancing, while "Deterministic" leverages orientation but not the uncertainty. "Task Error U." refers to the distance-based uncertainty due to ambiguity in the task (Eq. 3.2), "MonoLoco++ U." refers to the instance-based uncertainty estimated by our MonoLoco++.

Method	Accuracy (%) ↑ [Recall (%) ↑]			
	Easy	*Mod.*	*Hard*	*All*
W/o Orientation	84.0 [95]	80.9 [75]	82.5 [33]	83.3 [75]
Deterministic	80.5 [95]	77.9 [75]	79.0 [33]	79.8 [75]
Task Error U.	84.2 [95]	81.4 [75]	85.3 [33]	83.6 [75]
MonoLoco++ U.	**84.7** [95]	**81.6** [77]	**85.3** [33]	**84.0** [75]

which accounts for occlusions or unusual locations. We compare this technique to (i) a baseline approach that leverages 3D localization but not orientation, (iI) a deterministic approach by only using the distance d, and (iii) a probabilistic approach where the uncertainty is provided by the task error defined in Eq. 3.2.

Results. Table 7.1 shows the results for the *talking* activity in the Collective Activity Dataset [43]. Our MonoLoco++ detects whether people are talking from a single RGB image with 91.4% accuracy without being trained on this dataset, but only using the estimated 3D localization and orientation. The uncertainty estimation plays a crucial role in dealing with noisy 3D localizations as shown in the ablation study of Table 7.1. All approaches use the same values for 3D localization and orientation, but they differ in their uncertainty component. The biggest improvement is given from a deterministic approach to a probabilistic one. *Task Error U.* refers to the task error uncertainty of Eq. 3.2, which grows linearly with distance. *MonoLoco++ U.* refers to the estimated confidence interval from MonoLoco++, which are unique for each person. The role of uncertainty is also shown in Figures 7.4, and 7.5, where 3D localization errors are compensated by the voting procedure.

7.4.2 Social Distancing

Regarding social distancing, there are no fixed rules for evaluation. As previously discussed, the risk of contagion is higher when people are talking to each other [14], yet it may be necessary to maintain social distancing also when people are simply too close. Our goal is not to provide effective rules, but a framework to assess whether a given set of rules is respected.

Datasets. In the absence of a dataset for social distancing, we created one by augmenting 3D labels of KITTI dataset [64]. We apply Eq. 7.2 using the ground-truth localization and orientation to define whether people are violating social distancing. Once every person is assigned a binary attribute, we evaluate our accuracy on this classification task using our estimated 3D localization and orientation and applying the same set of rules.

Evaluation. We evaluate on the augmented KITTI dataset where every person has been assigned a binary attribute for social distancing. Coherently with the monocular 3D localization task, we evaluate on the val split of Chen *et al.* [39] even if no training is performed for this task. We use the same parameters as for the social interaction task, only relaxing the constraint on how people should look inward the o-space, and we set $R_{max} = 2 * r_{o-space}$. This corresponds to verifying whether both candidate centers μ_0, μ_1 are inside the o-space, as shown in Figure 7.3. The larger R_{max} in Eq. 7.2c, the more conservative the social distancing requirement. If Eq. 7.2c is removed completely, social distancing would only depend on the distance between people.

Results. Using the augmented KITTI dataset, we analyze whether social distancing is respected for 1760 people. Using the ground-truth localization and orientation we generate labels for which 36.8% of people do not comply with social distancing requirements. This is reasonable as KITTI dataset contains many crowded scenes. As shown in Table 7.2, our MonoLoco++ obtains an accuracy of 83.2%. We note that this dataset is more challenging than the Collective Activity one [43], as it includes people 40+ meters far as well as occluded instances. Qualitative results are shown in Figures 7.6 and 7.7, where our method estimates 3D localization and orientation, and verify social distancing compliance. In particular, Figure 7.7 shows that the network is able to accurately localize two overlapping people and recognize a potential risk of contagion, also based on people's relative orientation. In addition, we notice that orientation has a direct impact on reducing false alarms. Without orientation, the network estimates that 43% of instances violate social distancing requirements. Including orientation, the estimated number reaches 37%, almost on par with the ground-truth value of 38%.

7.5 Privacy

Our network analyzes 2D poses and does not require images to process the scene. In fact in Figures 7.1, 7.6 and 7.7, the original image is only shown to clarify the context, but is not processed directly by MonoLoco++. We leverage an off-the-shelf pose detector that could be embedded in the camera itself. We have designed our system to encourage a privacy-by-design policy [44], where images are processed internally by smart cameras [21] and only 2D poses are sent remotely to a secondary system. The 2D poses do not contain any sensible data but are informative enough to monitor social distancing.

We also note that smart cameras differentiate from other technologies by being non-invasive and mostly non-collaborative [44]. Unlike mobile applications, the user is not requested to share any personal data. On the contrary, a low-dimensional representation such as a set of semantic keypoints may prove helpful for privacy concerns.

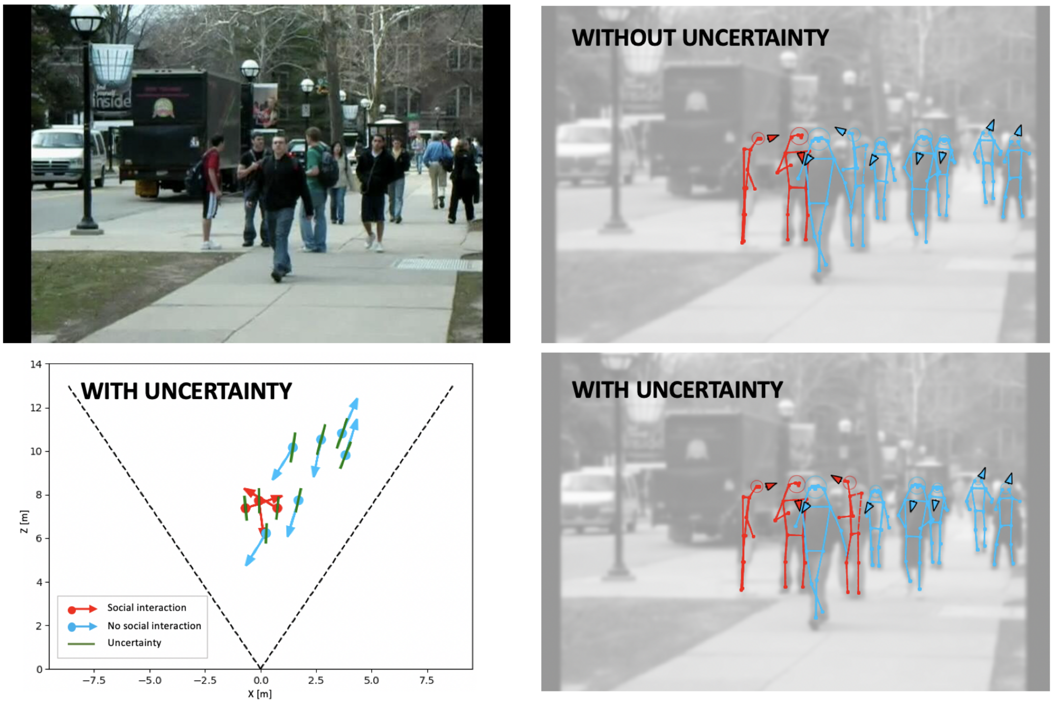


Figure 7.4 – Estimating whether people are talking to each other. The use of uncertainty makes the method more robust to 3D localization errors and improves the accuracy. The bird eye view shows the estimated 3D location and orientation of all the people. The color of the arrows indicates whether people are involved in talking.

7.6 Conclusion

While we have demonstrated the strengths of our method on popular tasks (monocular 3D localization and social interaction recognition), the COVID-19 outbreak has highlighted more than ever the need to perceive humans in 3D in the context of intelligent systems. We argued that to effectively monitoring social distancing, we should go beyond a measure of distance. Orientation and relative positions of people strongly influence the risk of contagion, and people talking to each other incur in higher risks than simply walking apart. Hence, we have presented an innovative approach to analyze social distancing, not only based on 3D localization but also on social cues. We hope our work will also contribute to the collective effort of preserving people's health while guaranteeing access to transportation hubs.

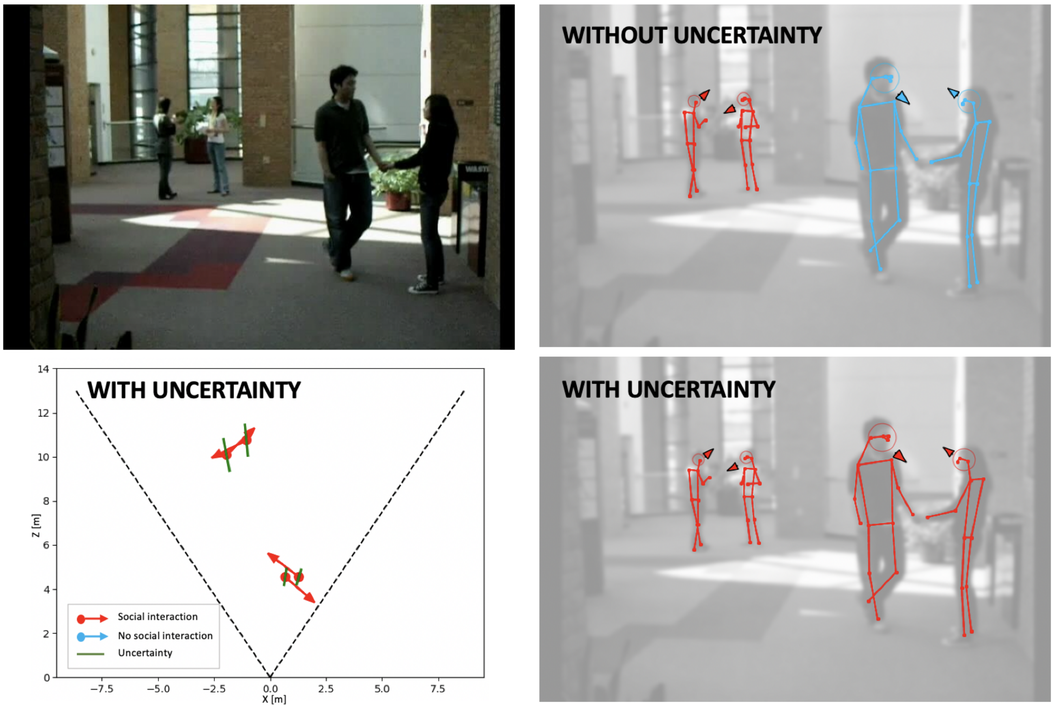


Figure 7.5 – Estimating whether people are talking. Even small errors in 3D localization can lead to wrong predictions. As shown in the bird eye view, the estimated location of the two people is only slightly off due to the height variation of the subjects. Uncertainty estimation compensates the error due to the ambiguity of the task.

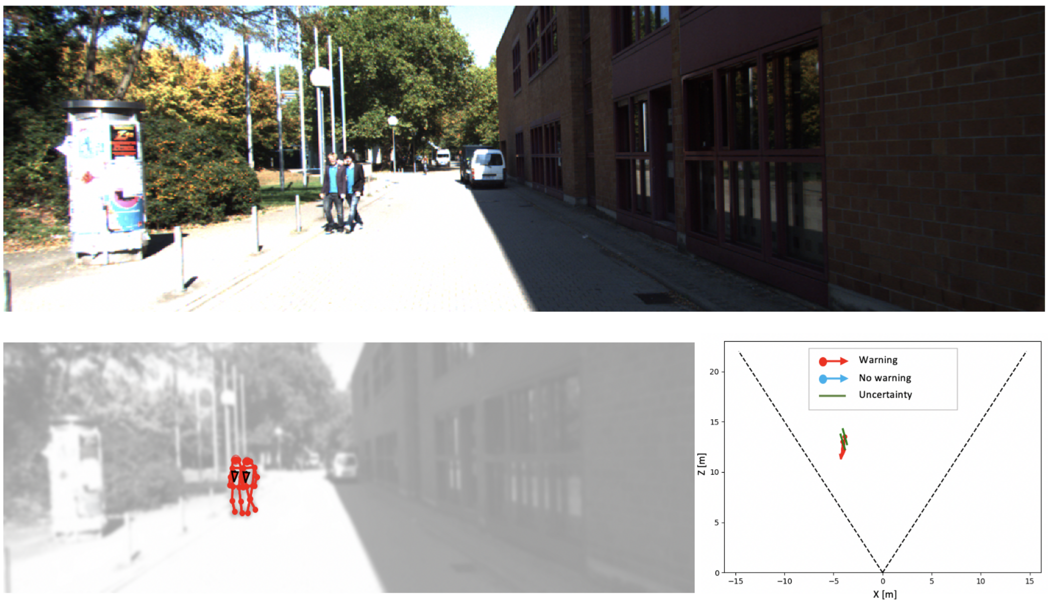


Figure 7.6 – 3D localization task. Illustration of two people walking and talking together. Our MonoLoco++ estimates 3D location, orientation and raises a warning when social distancing is not respected.

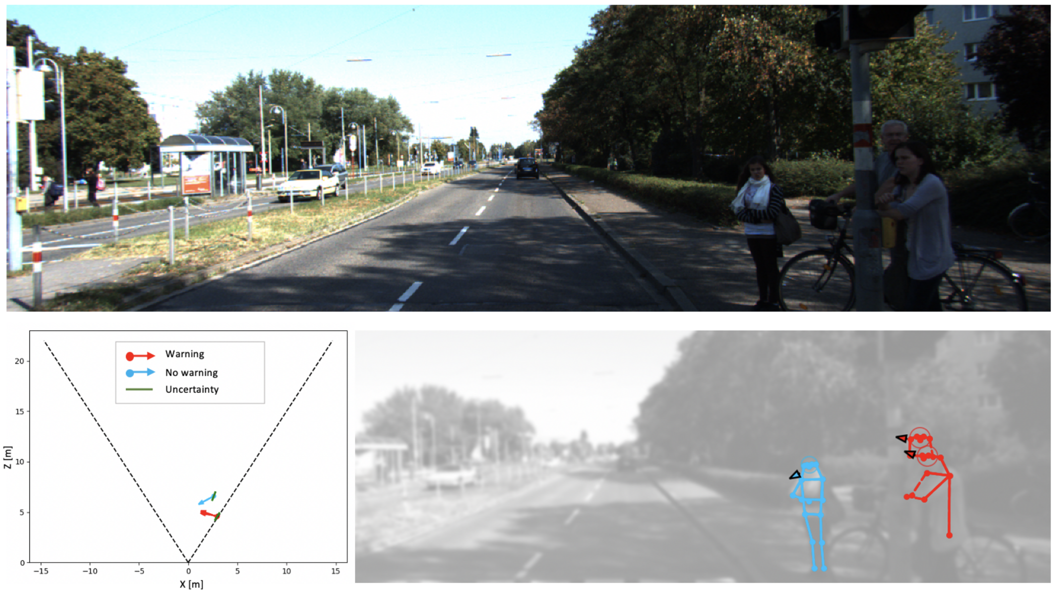


Figure 7.7 – Three people waiting at the traffic light. Two overlapping people are detected as very close to each other and the system warns for potential risk of contagion. A third person is located slightly more than two meters away and no warning is raised.

8 Conclusion

This thesis has examined the effectiveness of using semantic keypoints for locating humans in the 3D world and analyzing their behavior. We summarize the key results we have found, some of the limitations of our approach, and possible future extensions of it.

8.1 Findings

Chapter 2 introduced semantic keypoint and proposed a novel bottom-up pose detector based on composite fields. We found that our method based on composite fields is particularly suited for autonomous navigation settings: limited resolution on humans and high-density crowds where pedestrians occlude each other. Our method outperformed state-of-the-art bottom-up methods for human pose estimation while running an order of magnitude faster. We also showed how the concept of semantic keypoints is not limited to human pose estimation by extending it to car and animal poses and showing quantitative and qualitative results on multiple datasets [95, 94].

Chapter 3 set the basis for establishing the effectiveness of semantic keypoints for 3D pedestrian detection. Our proposed architecture achieved state-of-the-art results in monocular settings without access to any contextual information such as scene background or other objects locations. We also discovered that using a loss function based on the Laplace distribution is an effective technique to address the ambiguity in the task and predict calibrated confidence intervals. Our experiments have shown that the design is particularly well suited for small training data and cross-dataset generalization.

Chapter 4 built on top of the previous chapter by extending our architecture to multiple inputs from stereo cameras. We found that stereo-based disparity is highly effective only when explicitly combined with monocular cues. Hence, we developed an architecture (i) that jointly associates humans in left-right images, (ii) deals with occluded and distant cases in stereo settings by relying on the robustness of monocular cues, and (iii) tackles the intrinsic ambiguity of monocular perspective projection by exploiting prior knowledge of the human height distribution. Leveraging stereo and monocular information was crucial to improving performances and robustness to outliers. In addition, we observed that the official KITTI 3D metrics are not suited for the 3D pedestrian localization task and proposed a practical 3D metric tailored for humans.

Chapter 5 focused on autonomous driving applications of our method in two different settings. First, integrating our architecture in an end-to-end motion planner, we showed how a data-centric approach is actually more effective than a model-centric one to improve the intervention rate of autonomous vehicles due to pedestrian interactions. Second, we proved that our keypoints-based design is particularly effective in generalizing to unseen scenarios and, therefore, creating pseudo-labels of crowded scenes in London.

Chapter 6 went beyond locating humans in 3D, focusing on eye contact detection in real-world scenarios for autonomous vehicles with no control over the environment or the pedestrians' distance. To study domain adaptation, we created LOOK: a large-scale dataset for eye contact detection in the wild made of diverse and unconstrained scenarios for real-world generalization. We observed that a network based on semantic keypoints better estimates whether people are looking at the camera than end-to-end networks leveraging raw images. In addition, we explored the effectiveness of semantic keypoints to recognize various humans' actions in autonomous driving applications.

Chapter 7 concluded the body of our thesis by combining our studies on 3D localization and action recognition and by expanding our approach beyond the autonomous driving domain. We entered the field of proxemics by investigating how people use the space when interacting in groups and realized that, by extending our architecture with probabilistic rules, we could accurately detect F-formations and social interactions among humans. We also showed that analyzing social interactions is a more effective way to monitor social distancing and fight the COVID-19 outbreak. In addition, our proposed solution (i) is privacy-safe, (ii) works with any fixed or moving cameras, and (iii) does not rely on ground plane estimation.

In summary, experimental results have supported our initial belief: *Semantic keypoints are a powerful representation to perceive humans in the 3D world.*

8.2 Limitations and Future Work

Often, someone's greatest strength is also their main weakness. Similarly, semantic keypoints effectively highlight the important elements of an image but are at risk of neglecting other crucial components. For instance, understanding the surrounding scene is a prerequisite to discriminate whether a pedestrian is crossing the street at the zebra crossing or is jaywalking. Alternatively, in locating humans in 3D, the appearance may help indicate whether the detected person is a close-by child or a far-away adult.

Generalization of Semantic Keypoints. Our vision is to improve scene understanding for autonomous driving by extending the concept of semantic keypoints. In this work, we used them to encode the location in the image where certain information is available, but the information contained in a keypoint is not limited to parts of an object. It can represent the class of the object or a specific behaviour. In addition, while we have already expanded the concept of keypoints from humans to vehicles and animals in this thesis, we could go beyond dynamic objects and represent static ones, such as trees or zebra crossing. By modeling the entire scene with a meaningful low-dimensional representation, we could improve our visual understanding of the world while still filtering redundant information and, more importantly, leveraging the progress of 2D vision tasks to improve 3D perception. Our keypoints detector OpenPifPaf (described in Chapter 2) is particularly suited to extend the concept of semantic keypoints, as it can even detect a different number of joints for instances belonging to the same category. 3D information of any object can be obtained using nuScenes [31] or KITTI [64] datasets by projecting 3D point clouds into the image plane and applying a segmentation algorithm [187] to the image.

Modeling Object Relations with Semantic Keypoints. Another exciting extension of our work is to create an architecture that simultaneously analyzes all the instances in an image and exploits their relations to improve 3D object detection. We have already dealt with multiple inputs from stereo images in Chapter 4 and with human social interactions in Chapter 7. Tying together these works, we could expand the concept of relations combining:

- *Proximity relations. E.g.,* people close together should be predicted at similar depths,
- *Contextual relations. E.g.,* a person behind a tree is further away than the tree,
- *Identity relations. E.g.,* two instances in a pair of left-right images or consecutive temporal frames should be predicted at the same depth.

By modeling interactions between all objects in a scene, we have the potential to improve quantitative results and specifically tackle the long tail of 3D object detection. For example, in a group of people, the adults' predicted depth may help estimate the depth of the close-by children or occluded instances. For this application, the use of keypoints over raw images presents two advantages: (i) the number of input features is invariant to the size of the object and the depth in the scene, and (ii) keypoints are a convenient representation to perform data augmentation. Hence, we suggest augmenting 3D vision datasets with synthetic instances of outliers often missing in standard datasets, *e.g*., children close by their parents.
Moreover, we argue that one of the greatest setbacks to exploiting all types of relations in a scene is the unstructured input format: different images contain a different number of instances and, at test time, the number is not known in advance. Low-dimensional representations mitigate this issue and in Chapter 4 we have proposed a pairwise approach to deal with it. This approach, however, is limited in its scalability when moving from pairwise to multi-wise comparisons. Alternative neural network architectures, such as Graph Attention Networks [193], are worth to be explored as they offer the following advantages:

- they are parallelizable across node-neighbor pairs,
- they can be applied to graph nodes having different degrees,
- they are directly applicable to inductive learning problems, where the model has to generalize to unseen structures,

In conclusion, we encourage the vision community to expand the concept of semantic keypoints beyond a spatial attribute and to investigate the interconnections between dynamic and static objects in a scene. More generally, we hope future research will continue to emphasize vulnerable road users, such as pedestrians and cyclists, and explicitly address the long tail of 3D perception to contribute to the safety of our roads.

Bibliography

[1] George Adaimi, Sven Kreiss, and Alexandre Alahi. "Rethinking Person Re-Identification with Confidence". In: *arXiv preprint arXiv:1906.04692* (2019).

[2] Maedeh Aghaei, Mariella Dimiccoli, and Petia Radeva. "Towards social interaction detection in egocentric photo-streams". In: *Eighth International Conference on Machine Vision (ICMV 2015)*. Vol. 9875. International Society for Optics and Photonics. 2015, p. 987514.

[3] Emanuel Sánchez Aimar, Petia Radeva, and Mariella Dimiccoli. "Social Relation Recognition in Egocentric Photostreams". In: *The IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 3227–3231.

[4] Alexandre Alahi, Michel Bierlaire, and Murat Kunt. "Object Detection and Matching with Mobile Cameras Collaborating with Fixed Cameras". In: *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*. 2008.

[5] Alexandre Alahi, Michel Bierlaire, and Pierre Vandergheynst. "Robust real-time pedestrians detection in urban environments with low-resolution cameras". In: *Transportation research part C: emerging technologies* 39 (2014), pp. 113–128.

[6] Alexandre Alahi, Albert Haque, and Li Fei-Fei. "RGB-W: When vision meets wireless". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3289–3297.

[7] Alexandre Alahi, Laurent Jacques, Yannick Boursier, and Pierre Vandergheynst. "Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments". In: *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. IEEE. 2009, pp. 1–8.

[8] Alexandre Alahi, David Marimon, Michel Bierlaire, and Murat Kunt. "A master-slave approach for object detection and matching with fixed and mobile cameras". In: *2008 15th IEEE International Conference on Image Processing*. IEEE. 2008, pp. 1712–1715.

[9] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. "Freak: Fast retina keypoint". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2012, pp. 510–517.

[10] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. "Tracking millions of humans in crowded spaces". In: *Group and Crowd Behavior for Computer Vision*. Elsevier, 2017, pp. 115–135.

[11] Alexandre Alahi, Judson Wilson, Li Fei-Fei, and Silvio Savarese. "Unsupervised camera localization in crowded spaces". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2666–2673.

[12] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[13] Gianluca Antonini, Santiago Venegas Martinez, Michel Bierlaire, and Jean Philippe Thiran. "Behavioral priors for detection and tracking of pedestrians in video sequences". In: *International Journal of Computer Vision* 69.2 (2006), pp. 159–180.

[14] Sima Asadi, Anthony S Wexler, Christopher D Cappa, Santiago Barreda, Nicole M Bouvier, and William D Ristenpart. "Aerosol emission and superemission during human speech increase with voice loudness". In: *Scientific reports* 9.1 (2019), pp. 1–10.

[15] Timur Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. "Social scene understanding: End-to-end multi-person action localization and collective activity recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4315–4324.

[16] World Bank. *Protecting public transport from the coronavirus... and from financial collapse*. 2020.

[17] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst". In: *arXiv preprint arXiv:1812.03079* (2018).

[18] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *The European Conference on Computer Vision (ECCV)*. Springer. 2006, pp. 404–417.

[19] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs". In: *arXiv preprint arXiv:1907.05047* (2019).

[20] Loris Bazzani, Marco Cristani, Diego Tosato, Michela Farenzena, Giulia Paggetti, Gloria Menegaz, and Vittorio Murino. "Social interactions by visual focus of attention in a three-dimensional environment". In: *Expert Systems* 30.2 (2013), pp. 115–127.

[21] Ahmed Nabil Belbachir. *Smart cameras*. Vol. 2. Springer, 2010.

[22] Younes Belkada, Lorenzo Bertoni, Romain Caristan, Taylor Mordan, and Alexandre Alahi. "Do Pedestrians Pay Attention? Eye Contact Detection in the Wild". In: *arXiv preprint arXiv:2112.04212* (2021).

[23] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. "MonoLoco: Monocular 3D Pedestrian Localization and Uncertainty Estimation". In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[24] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. "Perceiving Humans: From Monocular 3D Localization to Social Distancing". In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–18. DOI: 10.1109/TITS.2021.3069376.

[25] Lorenzo Bertoni, Sven Kreiss, Taylor Mordan, and Alexandre Alahi. "Monstereo: When monocular and stereo meet at the tail of 3d human localization". In: *The IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5126–5132.

[26] Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. "Who Left the Dogs Out? 3D Animal Reconstruction with Expectation Maximization in the Loop". In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 195–211.

[27] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. "Creatures great and SMAL: Recovering the shape and motion of animals from video". In: *Asian Conference on Computer Vision (ACCV)*. Springer. 2018, pp. 3–19.

[28] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. "Weight Uncertainty in Neural Network". In: *the International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2015, pp. 1613–1622.

[29] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[30] Erin Bromage. *The Risks - Know Them - Avoid Them*. 2020.

[31] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. "nuscenes: A multimodal dataset for autonomous driving". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631.

[32] Carlos Caetano, François Brémond, and William Robson Schwartz. "Skeleton image representation for 3d action recognition based on tree structure and reference joints". In: *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE. 2019, pp. 16–23.

[33] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. "Cross-domain adaptation for animal pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 9498–9507.

[34] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser A Sheikh. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *IEEE transactions on pattern analysis and machine intelligence (TPAMI)* (2019).

[35] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1302–1310.

[36] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6299–6308.

[37] Sergio Casas, Abbas Sadat, and Raquel Urtasun. "MP3: A Unified Model to Map, Perceive, Predict and Plan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14403–14412.

[38] Florian Chabot, Mohamed Ali Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. "Deep MANTA: A Coarse-to-Fine Many-Task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR))*. 2017, pp. 1827–1836.

[39] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. "Monocular 3d object detection for autonomous driving". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2147–2156.

[40] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. "3d object proposals for accurate object class detection". In: *Advances in Neural Information Processing Systems*. 2015, pp. 424–432.

[41] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. "Multi-view 3d object detection network for autonomous driving". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1907–1915.

[42] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. "HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5386–5395.

[43] Wongun Choi, Khuram Shahid, and Silvio Savarese. "What are they doing?: Collective activity classification using spatio-temporal relationship among people". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE. 2009, pp. 1282–1289.

[44] M. Cristani, A. D. Bue, V. Murino, F. Setti, and A. Vinciarelli. "The Visual Social Distancing Problem". In: *IEEE Access* 8 (2020), pp. 126876–126886.

[45] Marco Cristani, Loris Bazzani, Giulia Paggetti, Andrea Fossati, Diego Tosato, Alessio Del Bue, Gloria Menegaz, and Vittorio Murino. "Social interaction discovery by statistical analysis of F-formations." In: *British Machine Vision Conference (BMVC)*. Vol. 2. 2011, p. 4.

[46] Marco Cristani, Giulia Paggetti, Alessandro Vinciarelli, Loris Bazzani, Gloria Menegaz, and Vittorio Murino. "Towards computational proxemics: Inferring social relations from interpersonal distances". In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE. 2011, pp. 290–297.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[48] Wenlong Deng, Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. “Joint Human Pose Estimation and Stereo 3D Localization”. In: *The International Conference on Robotics and Automation (ICRA)*. 2020.

[49] Armen Der Kiureghian and Ove Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural Safety* 31.2 (2009), pp. 105–112.

[50] Tom van Dijk and Guido CHE de Croon. “How do neural networks see depth in single images?” In: *The IEEE International Conference on Computer Vision (ICCV)* (2019).

[51] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. “Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles”. In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1906–1915.

[52] Rudolfs Drillis, Renato Contini, and Maurice Bluestein. *Body segment parameters*. New York University, School of Engineering and Science, 1969.

[53] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. “Incorporating second-order functional knowledge for better option pricing”. In: *Advances in neural information processing systems* 13 (2000), pp. 472–478.

[54] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. “The pascal visual object classes challenge: A retrospective”. In: *International journal of computer vision* 111.1 (2015), pp. 98–136.

[55] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. “Rmpe: Regional multi-person pose estimation”. In: *International Conference on Computer Vision (ICCV)*. 2017, pp. 2334–2343.

[56] Zhijie Fang and Antonio M López. “Intention Recognition of Pedestrians and Cyclists by 2D Pose Estimation”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019).

[57] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. “Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection”. In: *the IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 3266–3273.

[58] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. “RT-GENE: Real-time eye gaze estimation in natural environments”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 334–352.

[59] JV Freeman, TJ Cole, S Chinn, PRh Jones, EM White, and MA Preece. “Cross sectional stature and weight reference curves for the UK, 1990.” In: *Archives of disease in childhood* 73.1 (1995), pp. 17–24.

[60] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *the International Conference on Machine Learning*. 2016, pp. 1050–1059.

[61] Yarin Gal, Jiri Hron, and Alex Kendall. “Concrete dropout”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 3581–3590.

[62] David Gallup, Jan-Michael Frahm, Philippos Mordohai, and Marc Pollefeys. "Variable baseline/resolution stereo". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[63] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. "Actor-transformers for group activity recognition". In: *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 839–848.

[64] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013).

[65] Jordan Golson. *Tesla's new Autopilot will run in 'shadow mode' to prove that it's safer than human driving*. Ed. by The Verge. 2016. URL: https://www.theverge.com/2016/10/19/13341194/tesla-autopilot-shadow-mode-autonomous-regulations.

[66] Alex Graves. "Practical variational inference for neural networks". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2348–2356.

[67] Edward Twitchell Hall. *The hidden dimension*. Vol. 609. Garden City, NY: Doubleday, 1966.

[68] H. Han, M. Zhou, and Y. Zhang. "Can Virtual Samples Solve Small Sample Size Problem of KISSME in Pedestrian Re-Identification of Smart Transportation?" In: *IEEE Transactions on Intelligent Transportation Systems* 21.9 (2020), pp. 3766–3776. DOI: 10.1109/TITS.2019.2933509.

[69] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" In: *The IEEE conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6546–6555.

[70] Joko Hariyono, Ajmal Shahbaz, Laksono Kurnianggoro, and Kang-Hyun Jo. "Estimation of collision risk for improving driver's safety". In: *Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE. 2016, pp. 901–906.

[71] Jeffrey Hawke, E Haibo, Vijay Badrinarayanan, Alex Kendall, et al. "Reimagining an autonomous vehicle". In: *arXiv preprint arXiv:2108.05805* (2021).

[72] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN". In: *The IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[74] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[75] Daniel Holland-Letz, Benedikt Kloss, Matthias Kässer, and Thibaut Müller. *Start me up: Where mobility investments are going*. Tech. rep. McKinsey & Company, 2019.

[76] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeff Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. “FIERY: Future Instance Prediction in Bird’s-Eye View from Surround Monocular Cameras”. In: *International Conference of Computer Vision (ICCV))*. 2021.

[77] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. “Joint Monocular 3D Vehicle Detection and Tracking”. In: *The IEEE International Conference on Computer Vision (ICCV)*. 2019.

[78] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks”. In: *The Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 1647–1655.

[79] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[80] Ashesh Jain, Luca Del Pero, Hugo Grimmett, and Peter Ondruska. “Autonomy 2.0: Why is self-driving always 5 years away?” In: *arXiv preprint arXiv:2107.08142* (2021).

[81] Hanbyul Joo, Tomas Simon, Mina Cikara, and Yaser Sheikh. “Towards Social Artificial Intelligence: Nonverbal Social Signal Prediction in A Triadic Interaction”. In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10873–10883.

[82] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. “End-to-end recovery of human shape and pose”. In: *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7122–7131.

[83] Piyawan Kasemsuppakorn and Hassan A Karimi. “A pedestrian network construction algorithm based on multiple GPS traces”. In: *Transportation research part C: emerging technologies* 26 (2013), pp. 285–300.

[84] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. “GSNet: Joint Vehicle Pose and Shape Reconstruction with Geometrical and Scene-aware Supervision”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 515–532.

[85] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in Neural Information Processing Systems*. 2017, pp. 5574–5584.

[86] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *The IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. 2018, pp. 7482–7491.

[87] Adam Kendon. *Conducting interaction: Patterns of behavior in focused encounters*. Vol. 7. CUP Archive, 1990.

[88] R. Kesten et al. *Lyft Level 5 AV Dataset 2019*. urlhttps://level5.lyft.com/dataset/. 2019.

[89] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[90] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. "Multiposenet: Fast multi-person pose estimation using pose residual network". In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 417–433.

[91] Julian Francisco Pieter Kooij, Nicolas Schneider, Fabian Flohr, and Dariu M Gavrila. "Context-based pedestrian path prediction". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 618–633.

[92] Parth Kothari, Sven Kreiss, and Alexandre Alahi. "Human trajectory forecasting in crowds: A deep learning perspective". In: *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* (2021).

[93] John Krafcik. *Waymo is opening its fully driverless service to the general public in Phoenix*. Ed. by Blog Post. 2020. URL: https://blog.waymo.com/2020/10/waymo-is-opening-its-fully-driverless.html.

[94] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. "OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association". In: *IEEE Transactions on Intelligent Transportation Systems* (2021).

[95] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. "Pifpaf: Composite fields for human pose estimation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11977–11986.

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems (NeurIPS)* 25 (2012), pp. 1097–1105.

[98] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. "Joint 3d proposal generation and object detection from view aggregation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–8.

[99] Jason Ku, Alex D Pon, and Steven L Waslander. "Monocular 3D Object Detection Leveraging Accurate Proposals and Shape Reconstruction". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11867–11876.

[100] Mikolaj E Kundegorski and Toby P Breckon. "A photogrammetric approach for real-time 3D localization and tracking of pedestrians in monocular infrared imagery". In: *SPIE Optics and Photonics for Counterterrorism, Crime Fighting, and Defence*. Vol. 9253. 2014.

[101] Abhijit Kundu, Yin Li, and James M. Rehg. "3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 3559–3568.

[102] Hideaki Kuzuoka, Yuya Suzuki, Jun Yamashita, and Keiichi Yamazaki. "Reconfiguring spatial formation arrangement by robot body orientation". In: *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2010, pp. 285–292.

[103] Joon-Young Kwak, Byoung Chul Ko, and Jae-Yeal Nam. "Pedestrian intention prediction based on dynamic fuzzy automata for vehicle driving at nighttime". In: *Infrared Physics & Technology* 81 (2017), pp. 41–51.

[104] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6402–6413.

[105] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10863–10872.

[106] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. "Stereo r-cnn based 3d object detection for autonomous driving". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 7644–7652.

[107] Peiliang Li, Tong Qin, et al. "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving". In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 646–661.

[108] Shuyuan Li, Jianguo Li, Hanlin Tang, Rui Qian, and Weiyao Lin. "ATRW: A Benchmark for Amur Tiger Re-Identification in the Wild". In: *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2590–2598. ISBN: 9781450379885. DOI: 10.1145/3394171.3413569.

[109] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. "Deep continuous fusion for multi-sensor 3d object detection". In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 641–656.

[110] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision (ICCV)*. 2017, pp. 2980–2988.

[111] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common Objects in Context". In: *The European Conference on Computer Vision (ECCV)*. 2014.

[112] Zechen Liu, Zizhang Wu, and Roland Toth. "SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.

[113] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. "SMPL: A skinned multi-person linear model". In: *ACM transactions on graphics (TOG)* 34.6 (2015), pp. 1–16.

[114] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. “SMPL: A Skinned Multi-Person Linear Model”. In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[115] Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. “Learning high-speed flight in the wild”. In: *Science Robotics* 6.59 (2021), eabg5810.

[116] Wayve Ltd. *Emerging Behaviour of our Driving Intelligence with End to End Deep Learning*. Ed. by Blog Post. 2021. URL: https://wayve.ai/blog/driving-intelligence-with-end-to-end-deep-learning/.

[117] Wayve Ltd. *Unlocking Markets Faster: Building AVs that Generalise*. Ed. by Blog Post. 2021. URL: https://wayve.ai/blog/unlocking-markets-faster-building-avs-that-generalise/.

[118] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. “Shufflenet v2: Practical guidelines for efficient cnn architecture design”. In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 116–131.

[119] Srikanth Malla, Behzad Dariush, and Chiho Choi. “Titan: Future forecast using action priors”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11186–11196.

[120] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. “Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2069–2078.

[121] Roberto Martin-Martin*, Mihir Patel*, Hamid Rezatofighi*, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. “JRDB: A Dataset and Benchmark of Egocentric Robot Visual Perception of Humans in Built Environments”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2021).

[122] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. “A simple yet effective baseline for 3d human pose estimation”. In: *The IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 2659–2668.

[123] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. *DeepLabCut: markerless pose estimation of user-defined body parts with deep learning*. Tech. rep. Nature Publishing Group, 2018.

[124] Alexander Mathis, Mert Yüksekgönül, Byron Rogers, Matthias Bethge, and Mackenzie W Mathis. “Pretraining boosts out-of-domain robustness for pose estimation”. In: *Proceeding of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021.

[125] Damien Matti, Hazım Kemal Ekenel, and Jean-Philippe Thiran. “Combining LiDAR space clustering and convolutional neural networks for pedestrian detection”. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2017, pp. 1–6.

[126] Taylor Mordan, Matthieu Cord, Patrick Pérez, and Alexandre Alahi. "Detecting 32 Pedestrian Attributes for Autonomous Vehicles". In: *arXiv preprint arXiv:2012.02647* (2020).

[127] Francesc Moreno-Noguer. "3D Human Pose Estimation from a Single Image via Distance Matrix Regression". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1561–1570.

[128] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. "3d bounding box estimation using deep learning and geometry". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 7074–7082.

[129] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. "Learning from Synthetic Animals". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12386–12395.

[130] Jishnu Mukhoti and Yarin Gal. "Evaluating Bayesian Deep Learning Methods for Semantic Segmentation". In: *arXiv preprint arXiv:1811.12709* (2018).

[131] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *International Conference on Machine Learning ICML)*. 2010.

[132] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[133] Yurii E Nesterov. "A method for solving the convex programming problem with convergence rate O (1/kˆ 2)". In: *Soviet Mathematics Doklady*. Vol. 269. 1983, pp. 543–547.

[134] Yurrii Nesterov. "A method of solving a convex programming problem with convergence rate O(1/k2)". In: *Soviet Mathematics Doklady*. Vol. 27. 2. 1983, pp. 372–376.

[135] Alejandro Newell, Zhiao Huang, and Jia Deng. "Associative embedding: End-to-end learning for joint detection and grouping". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2277–2287.

[136] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 483–499.

[137] World Health Organization. *Infographics on Road Traffic Injuries*. Ed. by World Health Organization. 2021. URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[138] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model". In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 269–286.

[139] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Christoph Bregler, and Kevin P. Murphy. “Towards Accurate Multi-person Pose Estimation in the Wild”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 3711–3719.

[140] Seonwook Park, Adrian Spurr, and Otmar Hilliges. “Deep Pictorial Gaze Estimation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[141] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. “PyTorch: An imperative style, high-performance deep learning library”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 8024–8035.

[142] Talmo D Pereira, Diego E Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S-H Wang, Mala Murthy, and Joshua W Shaevitz. “Fast animal pose estimation using deep neural networks”. In: *Nature methods* 16.1 (2019), pp. 117–125.

[143] Dinh-Tan Pham, Quang-Tien Pham, Thi-Lan Le, and Hai Vu. “An Efficient Feature Fusion of Graph Convolutional Networks and Its Application for Real-Time Traffic Control Gestures Recognition”. In: *IEEE Access* 9 (2021), pp. 121930–121943.

[144] Jonah Philion and Sanja Fidler. “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 194–210.

[145] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation”. In: *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4929–4937.

[146] Boris T Polyak and Anatoli B Juditsky. “Acceleration of stochastic approximation by averaging”. In: *SIAM journal on control and optimization* 30.4 (1992), pp. 838–855.

[147] Alex D Pon, Jason Ku, Chengyao Li, and Steven L Waslander. “Object-centric stereo matching for 3d object detection”. In: *The International Conference on Robotics and Automation (ICRA)*. 2020.

[148] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. “Sampling-free Epistemic Uncertainty Estimation Using Approximated Variance Propagation”. In: *The IEEE International Conference of Computer Vision (ICCV)* (2019).

[149] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. “Frustum pointnets for 3d object detection from rgb-d data”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 918–927.

[150] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. “End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection”. In: *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 5881–5890.

[151] Zengyi Qin, Jinglu Wang, and Yan Lu. “Monogrnet: A geometric reasoning network for monocular 3d object localization”. In: *the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 8851–8858.

[152] Zengyi Qin, Jinglu Wang, and Yan Lu. “Triangulation Learning Network: from Monocular to Stereo 3D Object Detection”. In: *arXiv preprint arXiv:1906.01193* (2019).

[153] Akshay Rangesh and Mohan Manubhai Trivedi. “When vehicles see pedestrians with phones: A multicue framework for recognizing phone-based activities of pedestrians”. In: *IEEE Transactions on Intelligent Vehicles* 3.2 (2018), pp. 218–227.

[154] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. “PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 6262–6271.

[155] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. “Agreeing to cross: How drivers and pedestrians communicate”. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 264–269.

[156] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. “Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 206–213.

[157] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. “Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 206–213.

[158] Amir Rasouli and John K Tsotsos. “Autonomous vehicles that interact with pedestrians: A survey of theory and practice”. In: *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* 21.3 (2019), pp. 900–918.

[159] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. “Occlusion-net: 2d/3d occluded keypoint localization using graph networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 7326–7335.

[160] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection”. In: *the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[161] Patrick L Remington, William N Hall, Irving H Davis, Anita Herald, and Robert A Gunn. “Airborne transmission of measles in a physician’s office”. In: *Jama* 253.11 (1985), pp. 1574–1577.

[162] Michael D. Richard and Richard Lippmann. “Neural Network Classifiers Estimate Bayesian a posteriori Probabilities”. In: *Neural Computation* 3 (1991), pp. 461–483.

[163] Thomas Roddick, Alex Kendall, and Roberto Cipolla. “Orthographic Feature Transform for Monocular 3D Object Detection”. In: *the British Machine Vision Conference (BMVC)*. 2019.

[164] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. "LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images". In: *IEEE transactions on pattern analysis and machine intelligence* (2019).

[165] T. Rowntree, C. Pontecorvo, and I. Reid. "Real-Time Human Gaze Estimation". In: *2019 Digital Image Computing: Techniques and Applications (DICTA)*. 2019, pp. 1–7. DOI: 10.1109/DICTA47822.2019.8945919.

[166] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International Conference on Computer Vision (ICCV)*. Ieee. 2011, pp. 2564–2571.

[167] Rudolph J Rummel. "Understanding conflict and war: Vol. 5: The Just Peace". In: *Beverly Hills, California: Sage Publications* (1981).

[168] David Ruppert. *Efficient estimations from a slowly convergent Robbins-Monro process*. Tech. rep. Cornell University Operations Research and Industrial Engineering, 1988.

[169] Humberto Saenz, Huiming Sun, Lingtao Wu, Xuesong Zhou, and Hongkai Yu. "Detecting phone-related pedestrian distracted behaviours via a two-branch convolutional neural network". In: *IET Intelligent Transport Systems* 15.1 (2021), pp. 147–158.

[170] Tim Salimans, Diederik Kingma, and Max Welling. "Markov chain monte carlo and variational inference: Bridging the gap". In: *the International Conference on Machine Learning*. 2015, pp. 1218–1226.

[171] Héctor Corrales Sánchez, Antonio Hernández Martínez, Rubén Izquierdo Gonzalo, Noelia Hernández Parra, Ignacio Parra Alonso, and David Fernandez-Llorca. "Simple Baseline for Vehicle Pose Estimation: Experimental Validation". In: *IEEE Access* 8 (2020), pp. 132539–132550.

[172] Stephen Se and Michael Brady. "Ground plane estimation, error analysis and applications". In: *Robotics and Autonomous systems* 39.2 (2002), pp. 59–71.

[173] João Sedoc, Anna Rogers, Anna Rumshisky, and Shabnam Tafreshi. "Proceedings of the Second Workshop on Insights from Negative Results in NLP". In: *Proceedings of the Second Workshop on Insights from Negative Results in NLP*. 2021.

[174] Francesco Setti, Chris Russell, Chiara Bassetti, and Marco Cristani. "F-formation detection: Individuating free-standing conversational groups in images". In: *PloS one* 10.5 (2015), e0123783.

[175] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9532–9545.

[176] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1874–1883.

[177] Jamie Shotton, Andrew Blake, and Roberto Cipolla. “Contour-based learning for object detection”. In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. Vol. 1. IEEE. 2005, pp. 503–510.

[178] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. “Efficient human pose estimation from single depth images”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2012), pp. 2821–2840.

[179] Mel Siegel. “The sense-think-act paradigm revisited”. In: *1st International Workshop on Robotic Sensing, 2003. ROSE’03*. IEEE. 2003, 5–pp.

[180] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez Antequera, and Peter Kontschieder. “Disentangling Monocular 3D Object Detection: From Single to Multi-Class Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[181] B.A. Smith, Q. Yin, S.K. Feiner, and S.K. Nayar. “Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction”. In: *ACM Symposium on User Interface Software and Technology (UIST)*. Oct. 2013, pp. 271–280.

[182] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. “Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5452–5462.

[183] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[184] Valentyn Stadnytskyi, Christina E Bax, Adriaan Bax, and Philip Anfinrud. “The airborne lifetime of small speech droplets and their potential importance in SARS-CoV-2 transmission”. In: *Proceedings of the National Academy of Sciences* (2020).

[185] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. “Which tasks should be learned together in multi-task learning?” In: *International Conference on Machine Learning ICML)*. PMLR. 2020, pp. 9120–9132.

[186] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. “Deep high-resolution representation learning for human pose estimation”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5693–5703.

[187] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. “Gated-SCNN: Gated Shape CNNs for Semantic Segmentation”. In: *The International Conference of Computer Vision (ICCV)*. 2019.

[188] Alexander Toshev and Christian Szegedy. “Deeppose: Human pose estimation via deep neural networks”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1653–1660.

[189] Khai N Tran, Apurva Bedagkar-Gala, Ioannis A Kakadiaris, and Shishir K Shah. "Social Cues in Group Formation and Local Interactions for Collective Activity Analysis." In: *International Conference on Computer Vision Theory and Applications (VISAPP*. 2013, pp. 539–548.

[190] U.S. Department of Transportation. *Automated Driving Systems 2.0, A Vision for Safety*. 2021. URL: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf.

[191] Dimitrios Varytimidis, Fernando Alonso-Fernandez, Boris Duran, and Cristofer Englund. "Action and intention recognition of pedestrians in urban traffic". In: *14th International Conference on Signal-Image Technology & Internet-based Systems (SITIS)*. IEEE. 2018, pp. 676–682.

[192] Sebastiano Vascon, Eyasu Zemene Mequanint, Marco Cristani, Hayley Hung, Marcello Pelillo, and Vittorio Murino. "A game-theoretic probabilistic approach for detecting conversational groups". In: *Asian Conference in Computer Vision (ACCV)*. Springer. 2014, pp. 658–675.

[193] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[194] Peter M Visscher. "Sizing up human height variation". In: *Nature genetics* 40.5 (2008), p. 489.

[195] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. "Glue: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[196] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. "FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection". In: *arXiv preprint arXiv:2104.10956*. 2021.

[197] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8445–8453.

[198] Waymo. *Waymo Open Dataset: An autonomous driving dataset*. 2019.

[199] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. "Convolutional Pose Machines". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4724–4732.

[200] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. "Deepsfm: Structure from motion via deep bundle adjustment". In: *European Conference on Computer Vision*. Springer. 2020, pp. 230–247.

[201] Julian Wiederer, Arij Bouazizi, Ulrich Kressel, and Vasileios Belagiannis. "Traffic Control Gesture Recognition for Autonomous Vehicles". In: *The International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 10676–10683.

[202] Sascha Wirges, Marcel Reith-Braun, Martin Lauer, and Christoph Stiller. "Capturing Object Detection Uncertainty in Multi-Layer Grid Maps". In: *arXiv preprint arXiv:1901.11284* (2019).

[203] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. "Data-driven 3D Voxel Patterns for object category recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1903–1911.

[204] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. "Subcategory-aware convolutional neural networks for object proposals and detection". In: *The IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2017, pp. 924–933.

[205] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. "Beyond pascal: A benchmark for 3d object detection in the wild". In: *Proceeding of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2014, pp. 75–82.

[206] Bin Xiao, Haiping Wu, and Yichen Wei. "Simple baselines for human pose estimation and tracking". In: *The European Conference on Computer Vision (ECCV)*. 2018, pp. 466–481.

[207] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[208] Bin Xu and Zhenzhong Chen. "Multi-Level Fusion Based 3D Object Detection From Monocular Images". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2345–2353.

[209] Yi Yang, Simon Baker, Anitha Kannan, and Deva Ramanan. "Recognizing proxemics in personal photos". In: *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 3522–3529.

[210] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. "Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving". In: *arXiv preprint arXiv:1906.06310* (2019).

[211] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. "Robust learning through cross-task consistency". In: *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11197–11206.

[212] Paul A Zandbergen. "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning". In: *Transactions in GIS* 13 (2009), pp. 5–25.

[213] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. "Deep network for the integrated 3d sensing of multiple people in natural images". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8410–8419.

[214] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. "ETH-XGaze: A Large Scale Dataset for Gaze Estimation under Extreme Head Pose and Gaze Variation". In: *European Conference on Computer Vision (ECCV)*. 2020.

[215] Yu Zhang and Qiang Yang. “An overview of multi-task learning”. In: *National Science Review* 5.1 (2018), pp. 30–43.

[216] Yilin Zhao. “Mobile phone location determination and its impact on intelligent transportation systems”. In: *IEEE Transactions on intelligent transportation systems* 1.1 (2000), pp. 55–64.

[217] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. “Scalable person re-identification: A benchmark”. In: *The IEEE international conference on computer vision*. 2015, pp. 1116–1124.

[218] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. “Objects as points”. In: *arXiv preprint arXiv:1904.07850* (2019).

[219] Yin Zhou and Oncel Tuzel. “Voxelnet: End-to-end learning for point cloud based 3d object detection”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4490–4499.

[220] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael Black. “Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images “In the Wild””. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 5358–5367.

[221] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. “Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3955–3963.

LORENZO BERTONI

I work on autonomous driving technology, and I have a passion for innovative products that can better serve society. I love to collaborate and move things forward, marrying creativity with strong analytical skills.

ACADEMIC EXPERIENCE

École Polytechnique Fédérale de Lausanne (EPFL) — **Lausanne, Switzerland**
Doctoral Assistant in the Visual Intelligence for Transportation (VITA) Lab — *2018 – 2022*

- Researcher on deep learning for autonomous driving with a focus on 2D/3D vision
- Co-author of seven publications in major machine learning conferences (CVPR, ICCV) and journals
- Developer of the 3D vision library MonoLoco and contributor of the 2D human pose estimator PifPaf
- Creator of a privacy-safe, open-source social distancing algorithm to fight COVID-19. The work has been featured in the BBC 4Tech show, EPFL news, and swiss newspapers
- Teaching assistant in *Computational Thinking* and *Deep Learning for Autonomous Vehicles* courses

Pi School of Artificial Intelligence — **Rome, Italy**
Summer School, Deep Learning — *2018*

- Kicked off a new collaboration with a customer to develop an AI-based business solution for advertising
- Created shared metrics with customers to monitor progress and gathered weekly feedbacks from customers to shape the product's direction

University of California, Berkeley — **Berkeley, USA**
Visiting Scholar, Model Predictive Control Lab — *2016*

- Programmed and designed an experimental Adaptive Cruise Control for autonomous electric vehicles, which performs a real-time trajectory optimization and minimizes the energy consumption

PROFESSIONAL EXPERIENCE

Wayve — **London, UK**
Driving Intelligence and Product Internship — *2021*

- Participated in the instantiation of Wayve’s Product function by helping the team translate between their Product visions and the language of ML research, while learning about strategic product planning
- Worked on redefining the interactions between self-driving cars and vulnerable road users, improving performances and interpretability

Oliver Wyman — **Milan, Italy**
Management Consultant — *2017*

- Worked on three projects for different companies, prioritizing tasks under strict time constraints, and proactively communicating with clients to ensure all stakeholders were aligned
- Working with a team of data scientists to deliver management interventions that generated investments, meeting milestones and deliverable dates.
- Led comprehensive risk analyses for a major Italian Bank, developing strategical documents addressed to the Chair of the Supervisory Board of the European Central Bank

EDUCATION

École Polytechnique Fédérale de Lausanne (EPFL) **Lausanne, Switzerland**
PhD Candidate in Electrical Engineering *2019 – 2022*

University of Illinois at Chicago **Chicago, USA**
MSc in Mechanical Engineering, focus on Automation – GPA 4.0/4.0 *2015 – 2017*

- Dissertation: "Ecological Cooperative Adaptive Cruise Control for Autonomous Electric Vehicles"

Politecnico di Torino **Turin, Italy**
Master Degree in Energy and Nuclear Engineering, 110/110 cum laude *2011 – 2016*
Bachelor Degree in Energy Engineering – 110/110 cum laude

VOLUNTEERING EXPERIENCE

Hospital la Croix **Cotonou, Benin**
Volunteer *2014*

- Helped in the malnourished children centre, training mothers on the best feeding habits
- Assisted in 10+ surgery operations, serving as an assistant of the Orthopaedic's Head Physician of the Cottolengo Hospital

PRIZES AND AWARDS

- **Pi Campus Venture Capital Fund:** 20.000€ grant to develop an AI solution for industry partners
- **Banca Sella and Fondazione G. Agnelli Honor Award:** 10.000€ prize to repay in full a student loan. Only one student per year obtains the award based on academic excellence
- **PoliTo Exchange Program Scholarship**: 6-months scholarship to pay for accommodation during the exchange program (1st out of 182 candidates based on merit)
- **TOP-UIC Scholarship**: 8000€ prize to defray expenses to top-ranked students admitted to an US college
- **Water-polo US National Championship 2015/2016**: 7th place achieved with the University of Illinois team

ADDITIONAL INFORMATION

- **Languages - English**: IELTS examination (proficient user), **Italian**: native, **French**: B1
- **Computer skills** – Python (PyTorch and TensorFlow), Matlab, C++, SAS, SQL, Latex, Microsoft Excel
- **Hobbies -** Voracious reader, sports enthusiast (cycling, surfing, skiing, swimming), mindfulness meditator