Safety Analysis of Deep Learning based 2D Pedestrian Detectors in the Context of Autonomous Driving in Urban Traffic

Master Thesis in Computer Science

by

Alen Smajić

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

to the

AI Systems Engineering Lab (AISEL)
Faculty of Computer Science and Mathematics
Johann Wolfgang Goethe-Universität Frankfurt am Main

in cooperation with

Volkswagen Commercial Vehicles
Volkswagen AG

Supervisors:
Supervisor: Prof. Dr. Visvanathan Ramesh
Co-Supervisor: Prof. Dr. Gemma Roig
Industrial Supervisor: Yasin Bayzidi

December 23, 2022

For my family.

Acknowledgements

I am extremely grateful to my supervisor and chair of the AI Systems Engineering Lab (AISEL)[1], Prof. Dr. Visvanathan Ramesh. I really appreciate the long discussions we had during my research and the valuable feedback that he always provided. His tremendous experience deeply influenced my way of thinking about how to approach designing safe AI-based systems. Furthermore, I could not have undertaken this journey without the support of the Volkswagen AG and the people I worked with on the KI Absicherung[2] project. Therefore, I would like to express my deepest gratitude to my industrial supervisor, Yasin Bayzidi, who supported me during my research and gave me the opportunity to be part of this project. He was a true mentor who taught me many valuable lessons that enriched my future career.

Special thanks to Dr. Michael Rammensee, whose lectures at the AI Systems Engineering Lab (AISEL) inspired me to specialize in the field of AI. I am also thankful to my co-supervisor, Prof. Dr. Gemma Roig, whose lectures on computer vision greatly sparked my interest in AI-based perception systems, which are also the main focus of this thesis.

Lastly, I would like to extend my sincere thanks to my friends and family, who supported and motivated me during my studies.

[1] Funded by the German Federal Ministry of Education and Research (BMBF) under funding number 01IS19062. Homepage: http://www.ccc.cs.uni-frankfurt.de/aisel-ai-systems-engineering-lab/

[2] Funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). Homepage: https://www.ki-absicherung-projekt.de/en/

Disclaimer

The results, opinions and conclusions of this thesis are not necessarily those of Volkswagen AG.

Erklärung zur Abschlussarbeit

gemäß § 34, Abs. 16 der Ordnung für den Masterstudiengang Informatik vom 17. Juni 2019

Hiermit erkläre ich

Smajic, Alen

(Nachname, Vorname)

Die vorliegende Arbeit habe ich selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst.

Ebenso bestätige ich, dass diese Arbeit nicht, auch nicht auszugsweise, für eine andere Prüfung oder Studienleistung verwendet wurde.

Zudem versichere ich, dass die von mir eingereichten schriftlichen gebundenen Versionen meiner Masterarbeit mit der eingereichten elektronischen Version meiner Masterarbeit übereinstimmen.

Frankfurt am Main, den
23. Dezember 2022

Unterschrift der/des Studierenden

Abstract

AI-based computer vision systems play a crucial role in the environment perception for autonomous driving. Although the development of self-driving systems has been pursued for multiple decades, it is only recently that breakthroughs in Deep Neural Networks (DNNs) have led to their widespread application in perception pipelines, which are getting more and more sophisticated. However, with this rising trend comes the need for a systematic safety analysis to evaluate the DNN's behavior in difficult scenarios as well as to identify the various factors that cause misbehavior in such systems. This work aims to deliver a crucial contribution to the lacking literature on the systematic analysis of Performance Limiting Factors (PLFs) for DNNs by investigating the task of pedestrian detection in urban traffic from a monocular camera mounted on an autonomous vehicle. To investigate the common factors that lead to DNN misbehavior, six commonly used state-of-the-art object detection architectures and three detection tasks are studied using a new large-scale synthetic dataset and a smaller real-world dataset for pedestrian detection. The systematic analysis includes 17 factors from the literature and four novel factors that are introduced as part of this work. Each of the 21 factors is assessed based on its influence on the detection performance and whether it can be considered a Performance Limiting Factor (PLF). In order to support the evaluation of the detection performance, a novel and task-oriented Pedestrian Detection Safety Metric (PDSM) is introduced, which is specifically designed to aid in the identification of individual factors that contribute to DNN failure. This work further introduces a training approach for F1-Score maximization whose purpose is to ensure that the DNNs are assessed at their highest performance. Moreover, a new occlusion estimation model is introduced to replace the missing pedestrian occlusion annotations in the real-world dataset. Based on a qualitative analysis of the correlation graphs that visualize the correlation between the PLFs and the detection performance, this study identified 16 of the initial 21 factors as being PLFs for DNNs out of which the *entropy*, the *occlusion ratio*, the *boundary edge strength*, and the *bounding box aspect ratio* turned out to be most severely affecting the detection performance. The findings of this study highlight some of the most serious shortcomings of current DNNs and pave the way for future research to address these issues.

Keywords: Safe AI, DNN Robustness, Performance Limiting Factors, Deep Learning, Object Detection, Pedestrian Detection, Autonomous Driving

Zusammenfassung

KI-basierte Computer-Vision-Systeme spielen eine entscheidende Rolle bei der Umgebungswahrnehmung für das autonome Fahren. Obwohl die Entwicklung selbstfahrender Systeme bereits seit mehreren Jahrzehnten vorangetrieben wird, konnten erst in jüngster Zeit bahnbrechende Durchbrüche in künstlichen neuronalen Netzen (KNN) dazu führen, dass Umgebungswahrnehmungssysteme immer ausgereifter wurden. Mit diesem Aufwärtstrend besteht jedoch ein Bedarf an einer systematischen Sicherheitsanalyse, um das Verhalten von KNN in schwierigen Szenarien zu bewerten und um die verschiedenen Faktoren zu identifizieren, die ein Fehlverhalten solcher Systeme verursachen. Diese Arbeit zielt darauf ab, einen entscheidenden Beitrag zur fehlenden Literatur über die systematische Analyse verschiedener leistungseinschränkender Faktoren (LF) für KNN zu liefern, indem die Aufgabe der Fußgängererkennung im Stadtverkehr von einer monokularen Kamera, die auf einem autonomen Fahrzeug montiert ist, untersucht wird. Um die gemeinsamen Faktoren, die zu einem Fehlverhalten führen, zu untersuchen, werden 6 häufig verwendete moderne Objekterkennungsarchitekturen sowie 3 verschiedene Erkennungsaufgaben analysiert, unter der Verwendung eines großen synthetischen Datensatzes und eines kleineren realen Datensatzes zur Fußgängererkennung. Die systematische Analyse umfasst 17 Faktoren aus der Literatur und 4 neue Faktoren, die als Teil dieser Arbeit eingeführt werden. Jeder der 21 Faktoren wird auf der Grundlage seines Einflusses auf die Erkennungsleistung bewertet und ob er als ein LF angesehen werden kann. Zur Unterstützung der Bewertung der Erkennungsleistung wird eine neuartige und aufgabenorientierte Fußgängererkennungs-Sicherheitsmetrik eingeführt, die speziell für die Identifizierung der einzelnen Faktoren entwickelt wurde, die zum Ausfall von KNN beitragen. Darüber hinaus wird in dieser Arbeit ein Trainingsansatz zur F1-Score Maximierung vorgestellt, welcher sicherstellen soll, dass die KNN mit ihrer maximalen Erkennungsleistung analysiert werden. Außerdem wird ein neuartiges Modell zur Verdeckungsschätzung eingeführt, um die fehlenden Fußgängerverdeckungsangaben für den realen Datensatz zu generieren. Auf der Grundlage einer qualitativen Analyse der Korrelationsgraphen, welche die Korrelation zwischen den LF und der Erkennungsleistung visualisieren, wurden in dieser Studie 16 der ursprünglich 21 Faktoren als LF für KNN identifiziert, von denen sich die *Entropie*, das *Verdeckungsverhältnis*, die *Grenzkantenstärke* und das *Seitenverhältnis der Bounding Box* als die Faktoren herausstellten, die die Erkennungsleistung am stärksten beeinflussen. Die Ergebnisse dieser Studie heben einige der wichtigsten Einschränkungen der derzeitigen KNN hervor und ebnen den Weg für künftige Forschungen zur Lösung dieser Probleme.

Contents

List of Figures

List of Tables

Acronyms

2D-IS	2D Instance Segmentation. 7, 36, 37, 41, 46, 47, 72
2D-KD	2D Keypoint Detection. 7, 9, 36, 41, 46, 47, 72
2D-OD	2D Object Detection. 6, 7, 13, 36, 37, 41, 46, 47, 72, 74
3D-OD	3D Object Detection. 11
AP	Average Precision. xv, 10, 46, 48
CNN	Convolutional Neural Network. 6, 12, 14, 15
COCO	Common Objects in Context. xv, 10, 46, 48
DNN	Deep Neural Network. v, 1, 2, 6, 12–17, 22, 32, 36, 40, 41, 50–72, 74, B-1
FN	False Negative. 9, 10, 12, 14, 22, 23, 49, 69
FP	False Positive. 9, 10, 15, 22, 23, 49, 69
FPS	Frames Per Second. xv, 48
ICF	Integral Channel Features. 12
IoU	Intersection over Union. ix, 10, 18, 19, 22, 46, 48, A-1
KI-A	KI Absicherung. x, xi, xiii–xvi, 19, 21, 29, 31, 33–40, 43, 44, 46–58, 60–66, 68–70, 72, 73, A-15, B-1
LAMR	Log Average Miss-Rate. xv, 10, 46, 48
MAE	Mean Absolute Error. 40
mAP	mean Average Precision. 10, 46
MSE	Mean Squared Error. 40
PASCAL VOC	PASCAL Visual Object Classes. xv, 10, 14, 46, 48

PDSM	Pedestrian Detection Safety Metric. v, x, xi, xiii–xv, 2, 17, 19–23, 30, 41–43, 46–49, 61, 71, 72, A-1
PLF	Performance Limiting Factor. v, xvi, 2, 6, 11–17, 22, 23, 33, 36, 41, 46, 47, 49–74, B-1
RMSE	Root Mean Squared Error. 40, 74
SAE	Society of Automotive Engineers. ix, 4, 5
SGD	Stochastic Gradient Descent. 44
SRTP	Safety Relevant True Positive. 22, 23, 49
TP	True Positive. 9, 10, 22, 23, 49

Chapter 1

Introduction

1.1 Motivation

Self-driving systems powered by AI are promising to revolutionize the field of mobility. Besides the potential to save many human lives, a successful deployment of autonomous vehicles would be a sustainable solution to reducing traffic congestion, reducing the number of traffic accidents, reducing energy consumption, and increasing the productivity of each individual as a result of reduced manual driving time [9]. Despite recent technological advancements, especially in the field of AI, fully autonomous driving in open environments has still not been realized because of the sheer complexity of developing an automated driving system that is able to adapt to unknown situations in real time. Another challenge is the fact that the majority of computer vision models, which are required for perception and forming informed decisions, still fall short of human perception and reasoning [10]. Mastering the task of environment perception is especially important because subsequent driving decisions rely heavily on perceived information, and failure to do so could result in catastrophic events and a decrease in public trust in such technologies. Therefore, it is no wonder that modern prototypes of self-driving cars are equipped with numerous sensors that are able to capture rich environmental information and support the modularization of various perception tasks. Much of the recent progress in environment perception can be attributed to the paradigm shift in replacing traditional algorithms from signal processing, tracking, and control theory with machine learning approaches, of which Deep Neural Networks (DNNs) are most prominent [11]. Even though DNNs represent the current state-of-the-art in most perception tasks, they come at the cost of poor interpretability and transparency in the decision making process, which is crucial for assessing the safety of use. The common practice of optimizing DNNs on a training dataset and evaluating their performance on a holdout dataset is insufficient when it comes to the deployment in safety-critical domains. Such use cases necessitate the development of systematic safety analysis procedures for identifying performance bottlenecks and conceptual limitations within the system architecture. This becomes especially apparent when looking at the task of pedestrian detection. As pedestrians represent one of the most vulnerable groups of road users that will be interacting with self-driving vehicles, it is of utmost importance to analyze and understand the conditions under which the detection of a pedestrian fails. Identifying the Performance Limiting Factors (PLFs) is the first step in building robust systems that are safe and trustworthy.

1.2 Objectives and Scope

While there are numerous works in the literature that address the effects of individual factors on the DNN's performance [12, 13, 14, 15, 16, 17], there is no thorough study dedicated towards a clear identification of several PLFs that could offer comparable results based on the severity of each factor. The main objective of this study is therefore to conduct a systematic safety analysis in order to identify the various factors that lead to DNN failure. These factors are throughout this work referred to as PLFs. The scope of this systematic safety analysis is within the use case of pedestrian detection in urban traffic from a monocular camera mounted on an autonomous car.

The goal of this work is to provide valuable insights into the behavior of DNNs under challenging scenarios and various edge cases in order to form a better understanding of their limitations and weak spots. Such insights are critical for ensuring AI safety because they provide valuable information for future system improvements and can provide guidance for future research within specific use cases. The findings and approaches discussed in this thesis contribute to the pioneering work of developing common safety analysis procedures for assessing the performance of AI-driven systems in safety-critical domains.

1.3 Thesis Outline

The rest of the thesis is structured as follows: Chapter 2 introduces the theoretical background, which is required for a thorough understanding of the upcoming topics. Specifically, it introduces the perception task in autonomous driving and the basic concepts of pedestrian detectors. Related works are discussed in Chapter 3. The related literature is divided into two sections based on whether the PLF studies conducted by these works address the use-case of pedestrian detection or whether they report on other DNN-based perception use-cases. The methodology used in this study is presented in Chapter 4. This includes the derivation of the employed Pedestrian Detection Safety Metric (PDSM) for assessing the detection performance and an overview of the 21 factors whose influence on the detection performance is studied. Chapter 5 includes a detailed description of the experimental setup, discussing important aspects like the used datasets and DNN architectures for pedestrian detection. It further introduces a progressive training approach for F1-Score maximization in order to analyze the DNNs at their highest detection performance. The results of the study are presented graphically in Chapter 6, where the effects of each individual factor are discussed. Finally, a summary based on the findings from Chapter 6 and a final conclusion, alongside suggestions for future work, are given in Chapter 7.

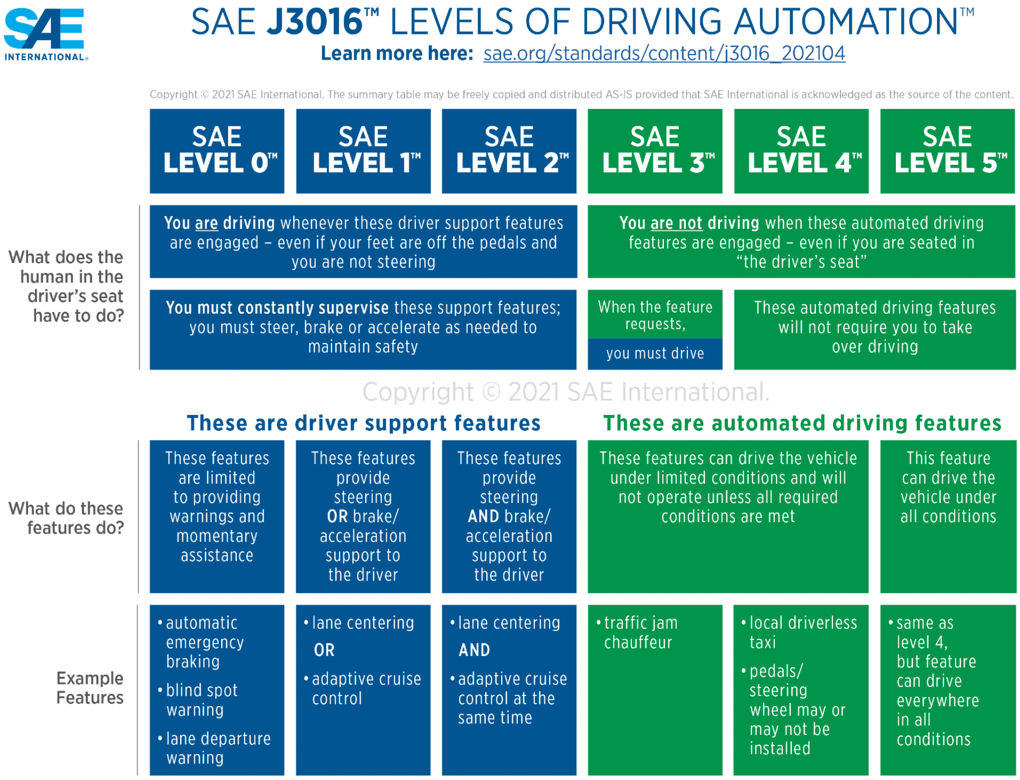
Chapter 2

Theoretical Background

2.1 Perception for Autonomous Driving

Autonomous driving is the act of utilizing a self-driving system that is capable of operating a vehicle without or with very little human intervention [18]. The Society of Automotive Engineers (SAE) defines five distinct levels of driving automation based on the capabilities of the utilized system, as shown in Figure 2.1. A fully functional self-driving system needs to master several difficult tasks that can be separated into three blocks, including the perception of the environment (aided by various sensors), the motion planning, and the act of controlling the vehicle [2]. Generally speaking, a self-driving car must be able to perceive, to plan, and to act in an almost infinite number of driving scenarios. Figure 2.2 visualizes the high-level architecture of a self-driving car as described by Shin *et al.* [2]. This section will further cover only aspects of the first block related to environment perception, which is also the main focus of this thesis.

Similar to how humans rely on visual information during the process of operating a car, the driving decisions made by a self-driving car heavily rely on information acquired by the environmental perception system. Because of this, self-driving cars are often equipped with a wide variety of sensors in order to capture as much environmental information as possible. Some of the utilized sensors are digital cameras, thermographic cameras, GPS, RADAR, LiDAR, SONAR, etc. Each sensor is responsible for capturing different sorts of information about the vehicle's surrounding area, which serves as input for the perception system responsible for scene understanding. Some of the important perception tasks include the detection and classification of other road users, traffic signs, traffic lights, lane markings, road obstacles, etc. Furthermore, objects that are in motion, including all road users, are being tracked to estimate their motion trajectory based on their past movement data. All of these environmental sensing tasks are often processed by individual systems or assemblies of such systems that have been engineered for each specific task. These systems frequently utilize imagery from several different sensors in a step called "sensor-fusion" to boost their performance by combining several sorts of sensor data.



SAE **J3016**™ LEVELS OF DRIVING AUTOMATION™

Learn more here: sae.org/standards/content/j3016_202104

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	**You are driving** whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			**You are not driving** when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			**These are automated driving features**		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering **OR** brake/ acceleration support to the driver	These features provide steering **AND** brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering **OR** • adaptive cruise control	• lane centering **AND** • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 2.1: SAE J3016 Levels of Driving Automation [1].

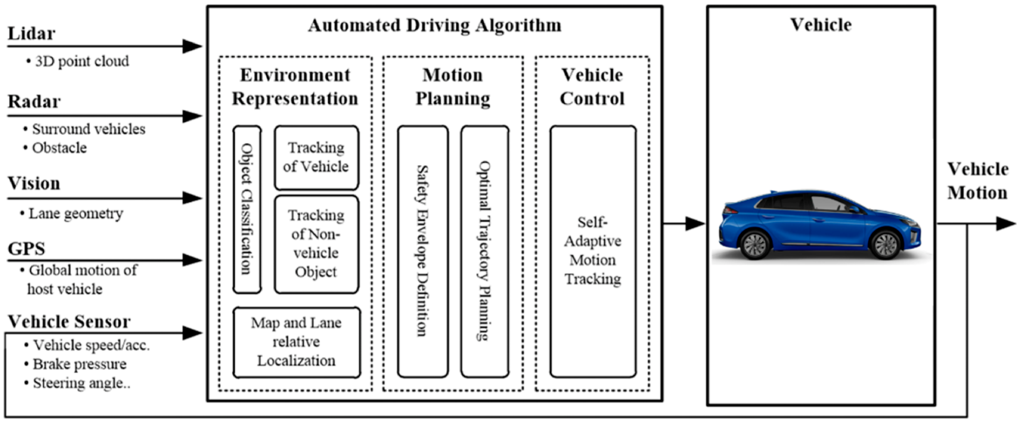


Figure 2.2: Overall system architecture of a self-driving car as described by Shin *et al.* [2]. The environment representation block uses commercialized sensors (LiDAR, RADAR, vision, etc.) to capture and then process all the environmental information. The motion planning block is used to plan the further driving behavior of the vehicle based on environmental information. The third block, being vehicle control, is responsible for controlling the steering and vehicle's physical motion with respect to the pregenerated motion plan [2].

Camera sensors provide rich information about the vehicle's environment by capturing high-resolution images that can be interpreted in ways that approximate human vision. They are also the only source of information when it comes to recognizing traffic signs and traffic lights, as well as extracting scene semantics by recognizing important visual features of other objects. Like all the other sensors, cameras also come with limitations since they are only able to capture visual information, and the image quality heavily relies on environmental factors like weather, for example [19]. Because of this, the perception systems that process this data are required to be engineered in a way that makes them robust against all the external factors that could potentially lead to system failure.

The study presented in this work focuses on the perception task of pedestrian detection based on visual information provided by a monocular camera sensor. The isolation of a single component of the perception system allows this study to better analyze and identify the external factors (PLFs) that decrease the detection performance of DNN-based 2D pedestrian detectors. Such valuable information can then be further utilized to improve the DNN's performance by making them more robust against such factors.

2.2 Fundamentals of 2D Pedestrian Detectors

This section introduces some fundamental terms and concepts of DNN-based 2D pedestrian detectors. DNNs are the current state-of-the-art machine learning approach for the task of object detection. They can be thought of as high-dimensional mathematical functions that have been optimized for specific tasks. Their unique ability is that for a given input of data (since this study is about perception, the inputs are images), which has not been shown to the DNNs before, they are able to produce an output, also known as predictions. In order to achieve this a DNN must be trained on data containing ground truth annotations for the given task. These DNNs are often referred to as "models". Furthermore, the term "deep learning" is often used when referring to the process of DNN training. The DNNs studied in this work belong to the group of Convolutional Neural Networks (CNNs).

2D Object Detection

The task of 2D Object Detection (2D-OD) involves the detection, localization, and classification of objects within a given input image. The systems that are utilized for 2D-OD can be either developed to detect a single class of objects, like pedestrians, for example, or they can be designed to handle multiple classes of different objects simultaneously. A simple 2D pedestrian detector, like the ones that are studied in this work, would therefore scan the input image to detect the presence of pedestrians and localize them. Such an input image consists of pixels

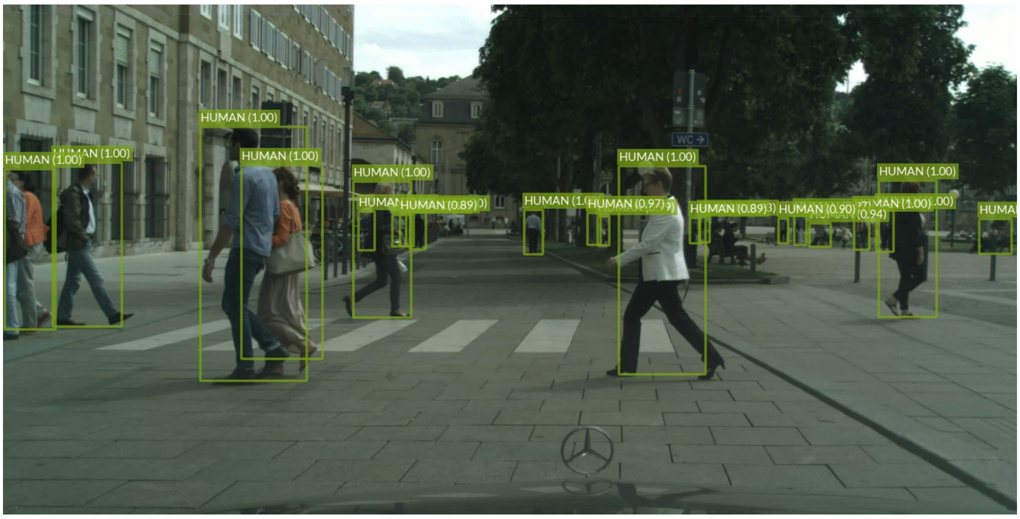


Figure 2.3: Sample image from the CityPersons dataset [3] with visualized pedestrian full-body bounding boxes. The pedestrian bounding boxes were extracted by one of the 2D pedestrian detectors that is studied in this work. The word human above each bounding box represents the name of the detected class, while the numbers represent the detector's confidence for each detection in the range from 0 to 1.

that are arranged in a rectangular grid, whose height and width correspond to the respective image dimensions. The localization of a pedestrian is represented by its respective bounding box information, which is usually given in the form of pixel coordinates that represent the upper left and lower right corners of the bounding box with respect to the 2D image grid. A proper bounding box should include the entire object while being as tight as possible. Figure 2.3 visualizes multiple bounding boxes on a given sample image from an urban driving scenario. All pedestrian detectors that are studied in this work are required to output full-body bounding boxes for the detected pedestrian instances. Furthermore, the task of 2D pedestrian detection (2D-OD) is extended in this study by the tasks of 2D Instance Segmentation (2D-IS) and 2D Keypoint Detection (2D-KD).

2D Instance Semegnetation

A pedestrian detection system that supports 2D-OD and 2D Instance Segmentation (2D-IS) will, in addition to the 2D bounding boxes, also segment the pixels within the bounding box. This means that each pixel within a bounding box will be assigned a binary class based on whether it represents the pedestrian instance or the background. This segmentation can then be visualized as a binary instance segmentation mask, since the resulting image can be used to mask out the image pixels that do not belong to the pedestrian instance. Figure 2.4 visualizes the

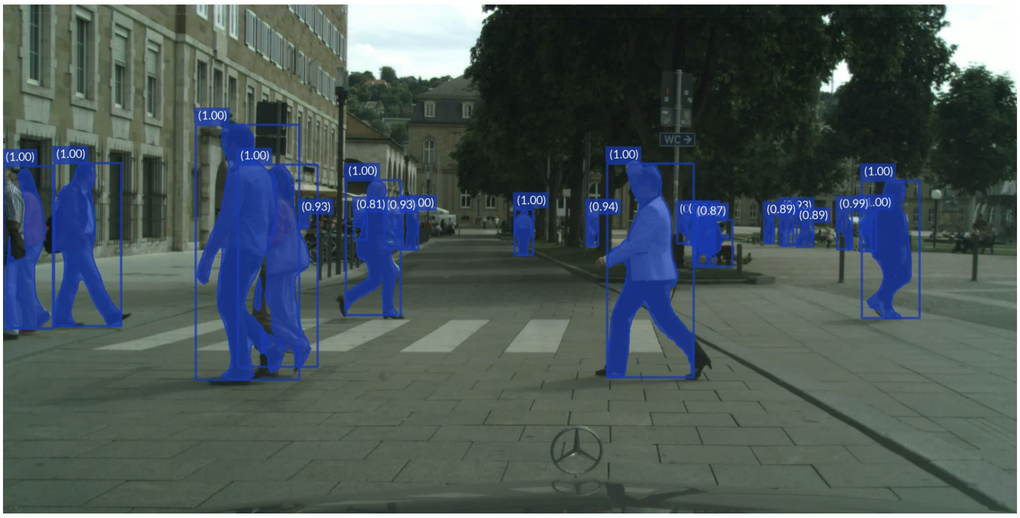


Figure 2.4: Sample image from the CityPersons dataset [3] with visualized pedestrian full-body bounding boxes and binary instance segmentation masks. The visualizations are based on the detection outputs of one of the 2D pedestrian detectors studied in this work. The numbers above each bounding box represent the detector's confidence for each detection in the range from 0 to 1.

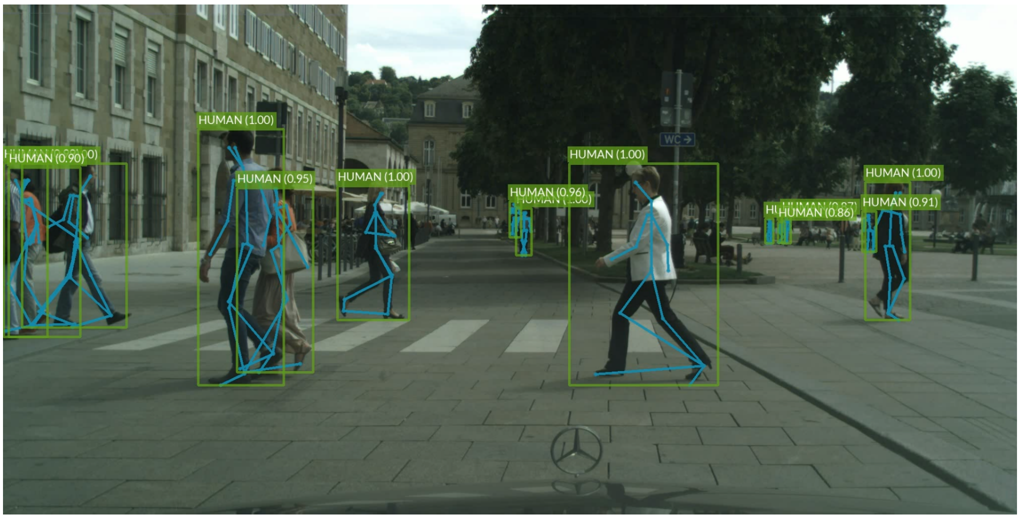


Figure 2.5: Sample image from the CityPersons dataset [3] with visualized pedestrian full-body bounding boxes and the respective keypoint skeletons. The visualizations are based on the detection outputs of one of the 2D pedestrian detectors studied in this work. The word human above each bounding box represents the name of the detected class, while the numbers represent the detector's confidence for each detection in the range from 0 to 1.

binary instance segmentation masks of several pedestrians in an urban driving scenario.

2D Keypoint Detection

Another detection task that is studied in this work is 2D-KD, which includes the detection, localization, and classification of pre-defined keypoints that belong to a single object. In the use case of pedestrian detection, a 2D-KD system would usually detect several human joints that form a keypoint skeleton. Such keypoint skeletons can then be further processed to approximate the human pose and its future movement trajectory. Each of the keypoints is represented by a single pixel coordinate with respect to the 2D image grid. Figure 2.5 visualizes the keypoint skeletons produced by a pedestrian detector in an urban driving scenario.

Detector Evaluation

The output of an object detector consists of a list of detected objects and the detector's confidence scores associated with each detection. In the first step, detection filtering is applied to remove all the detections whose confidence score is below the predefined **confidence threshold**, ranging between 0 and 1. In the next step, the filtered detections are compared to each other and matched with the predefined ground truth annotations. These ground truth annotations are generated manually by a human annotator and include bounding box information that is used to benchmark the system's detections. The process of matching detections with ground truth annotations will be discussed in the next paragraph. A True Positive (TP) represents a detection that has been successfully matched with an object's ground truth annotation. Since each detection can only be matched with a single ground truth annotation, several detections and ground truth annotations can remain unmatched. All unmatched detections represent False Positives (FPs), while all unmatched ground truth annotations represent False Negatives (FNs). In the example case of pedestrian detection, all pedestrian instances within a given input image that have been successfully detected are TPs, all detections that were unmatched and therefore do not represent a pedestrian detection are FPs and all pedestrian instances that remain undetected are FNs. Based on these evaluation outcomes, there are two standardized metrics that quantify the detection performance. The **precision** metric quantifies the ratio of TP detections to the total number of detections made, which is computed as the amount of TP detections divided by the sum of TP and FP detections. Its value can be interpreted as the average probability for a detection made by a detector to be a TP, hence it represents the detector's precision. **Recall** measures the degree to which the detector is able to recognize all of the object instances within a given image. Its value is simply the ratio of TP detections to the total number of ground truth annotations, which is computed as the amount of TP detections

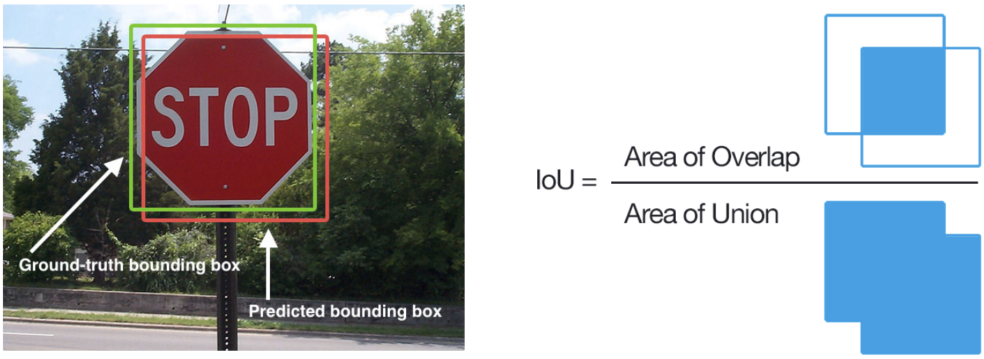


Figure 2.6: Visualization of the IoU measure on the use case of traffic sign detection [4]. The green bounding box represents the ground truth annotation, and the red bounding box corresponds to the predicted bounding box.

divided by the sum of TP and FN detections. The precision and recall metrics form the basis for other popular metrics that have been proposed in the literature, including the Average Precision (AP) as defined by the PASCAL Visual Object Classes (PASCAL VOC) [7], the Log Average Miss-Rate (LAMR) as defined by the Caltech pedestrian benchmark [8], and the mean Average Precision (mAP) as defined by Common Objects in Context (COCO) [6].

The matching of predicted bounding boxes and ground truth bounding boxes is computed based on the Intersection over Union (IoU) measure, which is responsible for quantifying the correctness of each detection. For a pair of predicted and ground truth bounding boxes, the resulting IoU can be computed as the fraction between the following values: The numerator is the pixel area of the inersection between the two bounding boxes. The denominator is the total pixel area of the union between the two bounding boxes. Figure 2.6 visualizes the IoU metric in the use case of traffic sign detection. A predicted bounding box is matched with a ground truth bounding box if the resulting IoU measure is above a predefined threshold, namely the **IoU threshold**. In the case that multiple predicted bounding boxes match the same ground truth annotation, the predicted bounding box with the highest IoU value is selected for matching. Once all predictions and ground truths have been processed, each of them has to be classified based on its evaluation outcome (TP, FP, or FN), as discussed in the previous paragraph.

Chapter 3

Related Work

3.1 Performance Limiting Factors (PLFs) for Pedestrian Detection

This section summarizes the related work on the topic of PLFs for pedestrian detection. These works include various studies that, in some form, investigated different kinds of factors and their influence on pedestrian detection performance. A more formal definition of these factors is given in section 4.2, where the concept of a Performance Limiting Factor (PLF) is derived.

Dollar *et al.* [8] introduced in 2011 the Caltech pedestrian benchmark. It consists of a large-scale real-world pedestrian detection dataset with over 350,000 pedestrian bounding boxes annotated within 250,000 image samples. However, due to the low image resolution (640x480), this dataset is rarely used nowadays. In their work, the authors investigated some of the dataset statistics, including the distribution of bounding box heights and bounding box aspect ratios. The authors also studied the most frequent occlusion patterns within the dataset and argued that the bottom part of the pedestrians is most frequently occluded. Finally, the authors stated that over 70% of all pedestrians appear occluded in at least one frame, which underlines the importance of the occlusion factor for the use-case of pedestrian detection.

In 2012, Geiger *et al.* [20] introduced the KITTI dataset for the tasks of stereo, optical flow, visual odometry/SLAM, and 3D Object Detection (3D-OD) within the use-case of autonomous driving. The object annotations also include pedestrians and cyclists. The authors studied some of the dataset properties, like the distribution of instances per object class and the distribution of occlusions. One particularity about the KITTI dataset is that its image samples have a much wider resolution[1] (1240x376) compared to other pedestrian detection datasets.

In 2015, Tian *et al.* [21] introduced *DeepParts*, which is a pedestrian detector consisting of extensive part detectors that are more robust towards occlusions.

[1] The KITTI dataset is not used within this study, since its image resolution highly deviates from the image resolution of the other datasets studied in this work.

The authors defined an extensive part pool and trained independent Convolutional Neural Networks (CNNs) for each part. The key advantage of this detector is the fact that, even highly occluded pedestrian instances contain visible body parts that can get detected by the respective part detectors, therefore improving the overall detection performance. The authors conducted experiments on the Caltech pedestrian benchmark and achieved state-of-the-art performance, outperforming the previous best method by 10%.

In 2016, Zhang *et al.* [22] analyzed the gap between the state-of-the-art pedestrian detectors at that time and the "perfect single frame detector", based on the Caltech pedestrian benchmark. The authors manually analyzed the detection errors for the best performing model and reported the two most severe PLFs to be the factor *small scale* and the factor *side view*, which marks pedestrians that appear from the side within the image. Further investigated PLFs include the *cyclist* factor and the *occlusion* factor. Moreover, the authors further investigated all undetected small-scale pedestrians to find the root cause for the DNN's failure. The authors hypothesized that low contrast and blurriness in small-scale pedestrian instances were the root causes of the poor detection performance. However, they reported that there is no correlation between low detection performance and low contrast or blurriness for the studied FNs.

In 2017, Mao *et al.* [23] studied what kinds of features could be added to the DNN-based pedestrian detectors to boost their performance. The authors experimented with various features that were added in the form of additional channels to the image data. The studied feature channels included an Integral Channel Features (ICF) [24] channel, an edge channel, a segmentation channel, a heatmap channel, an optical flow channel, and a disparity channel. The authors reported that the semantic channel increased detection performance at low resolution, while the ICF channel, the edge channel, and the heatmap channel increased localization accuracy at higher resolutions. Finally, the authors presented a novel framework for learning the aforementioned channel features as well as the task of pedestrian detection. This framework, named *HyperLearner*, was evaluated over several pedestrian detection datasets, in which competitive detection performance was achieved. In the same year, Wang *et al.* [25] studied the effects of pedestrian crowdedness on the detection performance. The authors experimented with a state-of-the-art pedestrian detector and demonstrated how pedestrian crowds lead to occlusions that result in reduced detection performance. To solve this problem, they propose a novel bounding box regression loss, named *repulsion loss*. Finally, the authors showed that the detectors trained by repulsion loss outperformed all the other state-of-the-art methods, which could most clearly be observed on the subset of occluded pedestrians. Later that year, Zhang *et al.* [3] introduced the CityPersons dataset, which is based on the CityScapes dataset [26] for semantic segmentation. This dataset will be discussed in more detail within section 5.1.2.

In 2019, von Bernuth *et al.* [14] argued that modern perception systems for automated driving are mostly trained on small-scale datasets that were taken under perfect weather conditions. To further robustify the DNNs, the authors proposed a novel augmentation framework for enhancing existing image samples with photo-realistic snow and fog effects. They compared their augmented images with real-world images containing fog and snow, demonstrating the effectiveness of their augmentation framework. Finally, the authors applied their method to a random subset of the KITTI dataset [27], which also contains pedestrian annotations for the task of 2D-OD. The authors used a benchmarked detector that performed well on the KITTI dataset and demonstrated that the detection performance suffers significantly with increasingly stronger weather influences from their augmentation framework. This study demonstrated the importance of severe weather conditions such as fog and snow on pedestrian detection performance. In the same year, Braun *et al.* [28] introduced the EuroCity Persons dataset[2] for pedestrian detection in urban traffic. This dataset contains over 238,200 pedestrian instances and over 47,300 image samples, which makes EuroCity Persons the currently largest real-world pedestrian detection dataset. It is also the first pedestrian detection dataset to introduce image samples taken at night. The authors optimized four state-of-the-art DNNs including Faster R-CNN [29], R-FCN [30], SSD [31], and YOLOv3 [32] and reported on Faster R-CNN achieving state-of-the-art performance. Furthermore, the authors reported that the diversity of the EuroCity Persons dataset leads to higher detection performance on other pedestrian detection datasets when applying transfer learning. With respect to the new nighttime factor, the authors reported that the detection performance is a few percentage points lower than at daytime.

In 2020, Xu *et al.* [33] developed a method for generating adversarial T-shirts to effectively evade person detectors in the physical world. The challenge in this task lay in the non-rigidity of the application surface (T-shirt), for which the authors had to develop a model of the temporal deformations that an adversarial T-shirt causes during pose changes and movement. They achieved strong attack performances in both digital and physical world tests, demonstrating the effectiveness of adversarial T-shirts on lowering the detection performance of DNNs.

In 2021, Lyssenko *et al.* [16] introduced so-called "relevance metrics", which are task-oriented performance measures. In their work, the authors studied the use case of pedestrian detection within the CARLA simulator [34] and the effects of the distance factor on the detection performance. Furthermore, the authors introduced a new dataset (comparable to the size of CityPersons [3]) based on the CARLA simulator with pedestrians at different distances ranging from 2 to 120 meters. They reported a linear decrease in detection performance with increasing distance, highlighting the importance of this factor. Based on this PLF analysis, the authors derived a metric that defines the highest distance up to which all

[2] Due to licensing issues, this dataset could not be used for this study.

pedestrians are detected. This metric could be used in the future to derive the detection range up to which AI safety is guaranteed.

More recently, in 2022, Hasan *et al.* [35] studied the generalization capabilities of modern DNN-based pedestrian detectors. They were able to show that recently proposed DNN architectures tailored towards pedestrian detection are biased towards the specific datasets for which they were designed. The authors conducted a cross-dataset evaluation and reported that the aforementioned models underperformed for even small domain shifts, while general object-detectors like Cascade R-CNN [36] without any "bells and whistles" were able to perform much better, demonstrating their ability to generalize over unseen data. Furthermore, the authors argued that modern pedestrian detection datasets still lack diversity, which is crucial for achieving detection performance comparable to a human. They propose a progressive fine-tuning strategy for improving the generalization capabilities of the studied DNNs by combining several pedestrian detection datasets. Finally, the authors conclude that, as of now, Convolutional Neural Networks (CNNs) outperform transformer-based DNNs, based on the results of their cross-dataset analysis.

In summary, the related work on PLFs for pedestrian detection has already identified some of the most sever factors that affect the detection performance. The occlusion factor is most frequently mentioned as this factor is responsible for most of the FNs. Although, there is promising work being conducted on the subject of PLFs for pedestrian detection, there is a clear gap in the literature on a systematic safety analysis of several PLFs in order to assess the severity of each factor leading to DNN failure.

3.2 Performance Limiting Factors (PLFs) for DNN-based Perception

This section summarizes the related work on the topic of PLFs for DNN-based perception, excluding the use-case of pedestrian detection, which has already been discussed in section 3.1.

In 2012, Hoiem *et al.* [12] conducted a large-scale PLF analysis on the PASCAL Visual Object Classes (PASCAL VOC) dataset [7] by investigating the effects of *occlusion*, *size*, *aspect ratio*, *visibility of parts*, *viewpoint*, *localization error*, and confusion with *semantically similar objects*, *other labeled objects*, and *background.* However, the detectors studied in this work are not DNN-based but rather comprise deformable parts models [37] and a cascade approach for object detection [38]. The authors report that the factors *size*, *localization error*, and confusion with *similar objects* are the most frequent forms of error.

In 2014, Luo *et al.* [39] introduced a mechanism to alleviate *adversarial attacks* within image classification by applying *foveation* to the input images of CNNs. These *adversarial attacks* are often executed by adding imperceptible changes to the image data, which leads to DNN failure, thus representing a serious safety concern.

In 2015, Nussberger *et al.* [40] studied the effects of lens flare for the use-case of aerial object tracking. Lens flare is the effect of scattered light in a lens system that produces flare artifacts in the image [41], which can potentially lead to DNN failure. The authors utilized the fact, that lens flare artifacts appear in form of a line close to the sun position in order to easier detect them. Furthermore, the authors combined the newly introduced lens flare filter with their aerial object tracking framework and reported that superior object tracking performance was achieved, due to the mitigation of FPs.

In 2019, Eykholt *et al.* [42] introduced *Robust Physical Perturbations (RP_2)*, a method for generating physical *adversarial stickers* that can be applied to individual objects. This method focuses specifically on the robustness of such attacks, since the perturbed objects are subjected to varying angles and distances from the viewing camera. RP_2 utilizes a two-stage optimization process in order to first localize the sensitive spots for placing the perturbations and then optimize the content of the perturbation sticker. In the same year, the authors published another work [43], extending their method towards object detection models. Later that year, Schneider *et al.* [44] studied the effects of image vignetting on the detection performance within the KITTI dataset [27]. The authors proposed a new approach for synthetic image augmentation by using a physics-based camera model. The ideal synthetic images used for training were further processed by the camera model to augment various optical effects, including image vignetting, which affects the pixel area around the image's borders, making them appear darker. The authors investigated several DNNs and their detection performance after training on ideal synthetic image samples and images with vignetting effects. The authors reported an increase in detection performance for the DNNs that were trained on the augmented image samples, effectively demonstrating the importance of the camera vignetting effect on DNN-based detection performance.

In 2021, Berghoff *et al.* [45] studied the topic of DNN robustness for the use-case of traffic sign recognition. The authors investigated several PLFs including *image noise*, *pixel perturbations*, *geometric transformations* and *colour transformations*. Furthermore, the authors introduced a robustness score, which measures the accuracy of the studied DNNs on various traffic signs and with respect to the aforementioned PLFs. Finally, the authors concluded that the robustness of the studied models at least partly correlates with the frequency of the PLFs values within the training dataset. This highlights the importance of the distribution of PLF values within the training splits. In the same year, Hess *et al.* [46] introduced a simulation framework for procedural world generation in order to conduct a systematic evaluation on continual learning for DNNs. This simulation framework supports various environmental factors including *il-*

lumination, *weather conditions*, *daytime and nighttime*, *color* and *surface reflections.* Furthermore, the framework allows for further configuration of the factors that are rendered within the simulation, providing a convenient way for generating image data with specific properties. Later that year, another simulation framework was introduced by Fischer *et al.* [47] for the use-case of traffic sign detection and classification. The simulation framework was inspired by the previous work [48] of one of the authors. Furthermore, the authors of this work also highlighted the importance of various environmental factors on the DNN's performance. Their simulation supports different *weather conditions*, *daytime*, *nighttime*, varying *sources of illumination*, and includes a framework for placing realistic sticker occlusion onto the traffic signs.

In 2022, the work from Bayzidi *et al.* [17] approached the topic of *Adversarial Attacks* from a different perspective. Motivated by the recent works in this field, the authors asked the question whether DNNs can be fooled by more realistic-looking stickers instead of the highly salient adversarial stickers. In their cross-analysis study, the authors investigated the effects of applying realistic stickers to the surface of traffic signs with the goal of deceiving state-of-the-art DNNs for image classification. The field test results revealed that the adversarial stickers from the literature have no effect on the DNNs in such physical scenarios because the distance between the camera and the traffic sign, as well as the respective camera angle, completely reduce the adversarial content of the stickers. Furthermore, the authors demonstrated superior misclassification performance (higher attack success rates) by carefully positioning the stickers on the right spots of the traffic sign to maximize the prediction loss of the DNNs. The results of this study highlight the serious safety concerns about the effects of physical sticker occlusion on detection performance. In the same year Pliushch *et al.* [49] investigated various image statistics and their correlation with the DNN's ability to learn. The authors studied several factors, including *edge strength*, *entropy*, and *segment count* and concluded that the order in which dataset instances are learned is highly independent of the individual DNNs and rather depends on the aforementioned image statistics.

The recent literature on PLFs for DNN-based perception mostly analyzed more general factors that are observable across all possible perception use-cases. Moreover, the related literature highlighted once more the importance of simulation frameworks and their usefulness for achieving higher robustness towards PLFs.

Chapter 4

Methodology

4.1 Pedestrian Detection Safety Metric (PDSM)

Since the scope of this study is within the use case of pedestrian detection for autonomous driving in urban traffic, there are several assumptions that can be used to reduce the noise during the experiments. Based on these assumptions, a novel and task-oriented Pedestrian Detection Safety Metric (PDSM) is derived in order to support the evaluation of the detection performance and to help identify the PLFs. A basic concept of PDSM is the distinction between safety-relevant and non-safety-relevant pedestrians that appear within a driving scene. This allows PDSM to focus more on the detection performance for safety-relevant individuals and the conditions under which the detection of a safety-relevant pedestrian fails.

4.1.1 Safety-Relevant and Non-Safety-Relevant Pedestrians

Based on the fact that the DNNs will be assessed in urban road scenarios, it can be assumed that the maximum driving speed will be kept at 50 km/h, as this is the standard speed limit in Germany and most EU countries [50]. TÜV Rheinland AG, a certified technical testing organization in Germany, estimates that at a speed of 50 km/h, the car's stopping distance in normal braking conditions is 40 meters and in emergency braking, 27.5 meters [51]. Based on these assumptions, the following definition is derived:

Definition 4.1. Any pedestrian instance that is more than 50 meters away from the autonomous vehicle is considered a non-safety-relevant pedestrian by PDSM.

The first argument for this distance threshold lies in the fact that a self-driving system must be able to detect pedestrians whose distance to the car is smaller than the car's stopping distance in order to adequately react to immediate danger and decide upon evasive maneuvers. Even though a potential collision with a pedestrian might not be avoidable, it is still of utmost importance that a self-driving system is able to detect and react, by adequately braking, for example. Aside from being able to detect pedestrians in its immediate vicinity, an autonomous vehicle must also be able to detect pedestrians that appear after the stopping distance up to a certain range in order to properly plan its driving behavior. For the scope of this study, the distance threshold has been set at 50 meters for urban areas, as this gives the self-driving system enough information to be able to drive autonomously. In analogy to this, all pedestrians that appear after 50 meters are considered to be non-safety-relevant. However, as the car is driving towards them, they will eventually be within the distance threshold and become safety-relevant for detection.

In many driving situations, at least two pedestrians will appear, frequently occluding each other. In such cases, it is often sufficient to detect the pedestrian who is staying in front, which means the one whose distance to the car is smaller. These kinds of pedestrians that are within the safety distance of 50 meters but are being occluded to a certain degree by other pedestrians are in this study referred to as "heavily crowded" pedestrians. One important property of heavily crowded pedestrians is that they appear as a crowd of pedestrian pixels in the 2D image and make it difficult for a detector to identify all instances. In the case where the pedestrians are physically located next to each other, it is sufficient that the system is aware of a pedestrian being located at that specific position. In the other case, where the pedestrians are physically farther away from each other but appear occluded because of the camera viewing angle, it can be assumed that as the car is moving, the camera viewing angle will adjust, making the heavily crowded pedestrian fully visible again. For the scope of this study, heavily crowded pedestrians are defined as follows:

Definition 4.2. The bounding box of a heavily crowded pedestrian has to overlap with another pedestrian's bounding box, and the overlap has to cover at least 60% of either one of the two bounding boxes. In such a scenario, the safety-relevant pedestrian, whose distance to the autonomous vehicle is smaller, will remain safety-relevant, while the overlapping pedestrian is regarded as heavily crowded and therefore non-safety-relevant for detection.

For a given bounding box, this can be simply computed by taking the pixel area of the overlap and dividing it by the total bounding box pixel area, as shown in the formula (4.13), which will be introduced in section 4.2. The overlap threshold of 60% has been chosen on a best effort basis after manual inspections. The reason for using the overlap measure instead of the standard IoU arises from the fact that IoU returns low values for bounding boxes with large size differences. In the case where a pedestrian in close proximity to the camera is partially occluding another pedestrian further away, depending on the severity of the occlusion, it would be preferable to mark the occluded pedestrian as heavily crowded and thus non-safety-relevant. As the distance between these pedestrians is high, the bounding box of the one next to the car will appear quite large compared to the occluded one. Even if the smaller bounding box is fully contained within the larger one, the IoU measure will still return low values as it takes the union area of the two bounding boxes into account. Because of these disadvantages, the overlap measure is used to determine whether or not a pedestrian bounding box is heavily crowded.

Figure 4.1 visualizes the distinction between safety-relevant and non-safety-relevant pedestrians in the form of a binary decision tree. Figures 4.2, 4.3, 4.4 and 4.5 depict two driving scenarios taken from the CityPersons [3] and the KI Absicherung (KI-A) [5] datasets, highlighting safety-relevant pedestrians, heavily crowded pedestrians (non-safety-relevant) and pedestrians whose distance to the camera exceed 50 meters (non-safety-relevant).

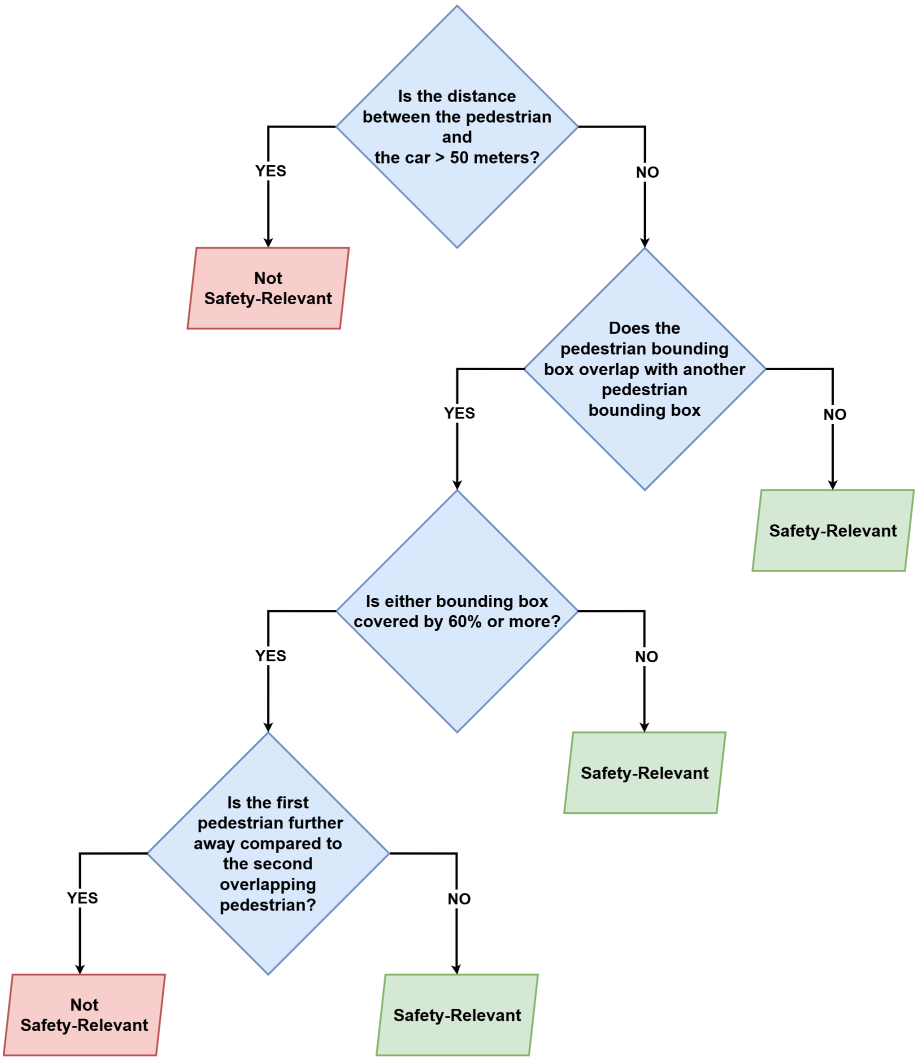

Is the distance between the pedestrian and the car > 50 meters?
YES
NO
Not Safety-Relevant
Does the pedestrian bounding box overlap with another pedestrian bounding box
YES
NO
Safety-Relevant
Is either bounding box covered by 60% or more?
YES
NO
Safety-Relevant
Is the first pedestrian further away compared to the second overlapping pedestrian?
YES
NO
Not Safety-Relevant
Safety-Relevant

Figure 4.1: Binary decision tree for classifying a pedestrian in relation to Pedestrian Detection Safety Metric (PDSM).

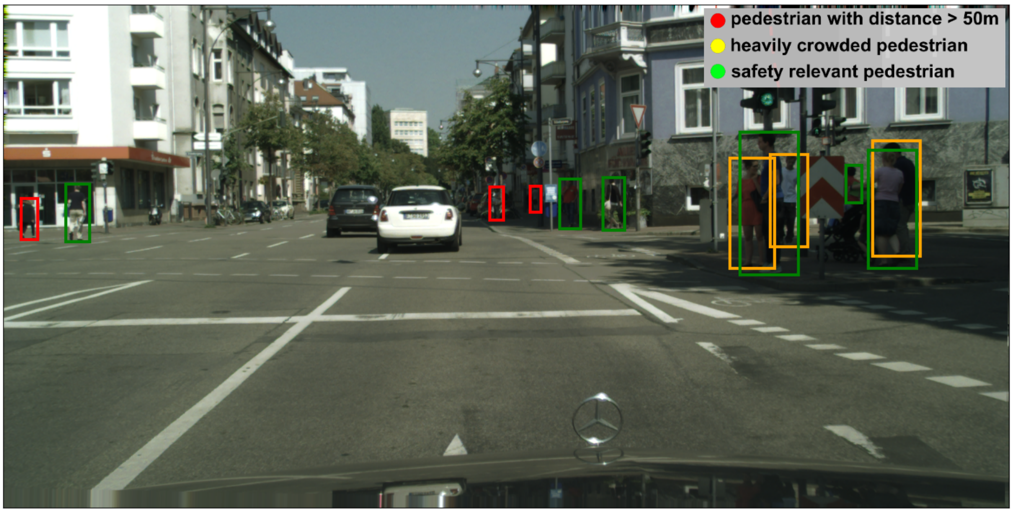

pedestrian with distance > 50m
heavily crowded pedestrian
safety relevant pedestrian

Figure 4.2: Visualization of pedestrian categories as defined by the Pedestrian Detection Safety Metric (PDSM) on a sample image from the CityPersons dataset [3].

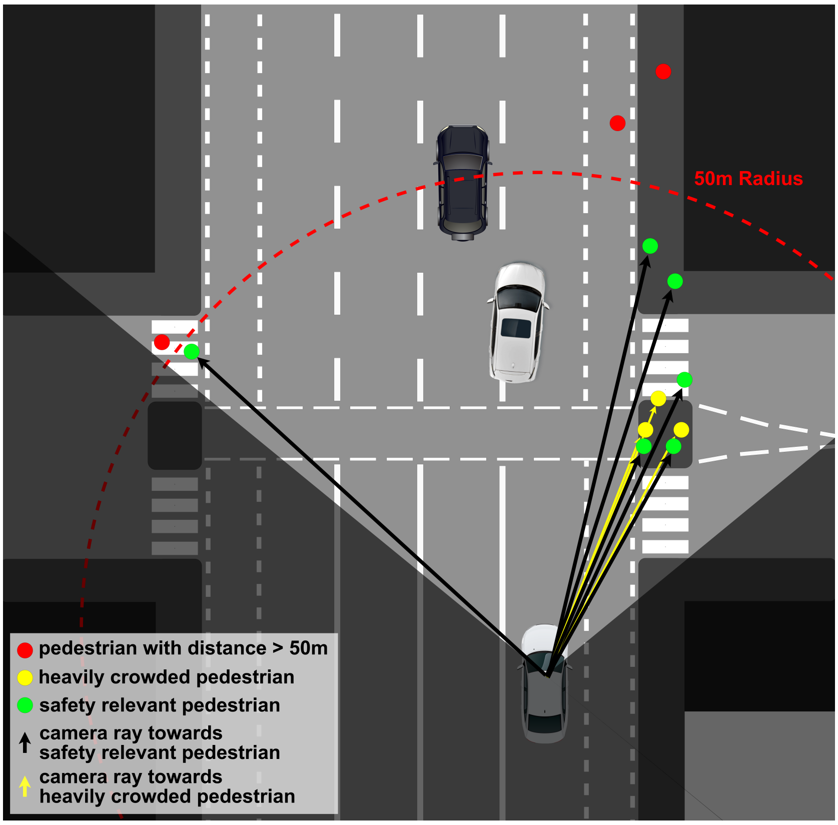

50m Radius
pedestrian with distance > 50m
heavily crowded pedestrian
safety relevant pedestrian
camera ray towards safety relevant pedestrian
camera ray towards heavily crowded pedestrian

Figure 4.3: A bird's-eye view of the driving scenario depicted in Figure 4.2, highlighting the 50-meter radius and the viewing rays of the camera angle.

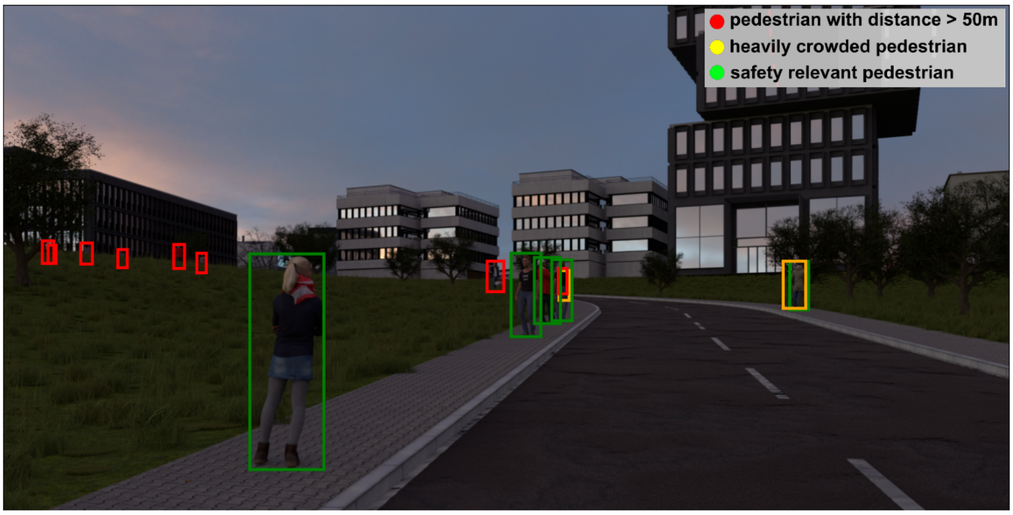

pedestrian with distance > 50m
heavily crowded pedestrian
safety relevant pedestrian

Figure 4.4: Visualization of pedestrian categories as defined by the Pedestrian Detection Safety Metric (PDSM) on a sample image from the KI Absicherung (KI-A) dataset [5].

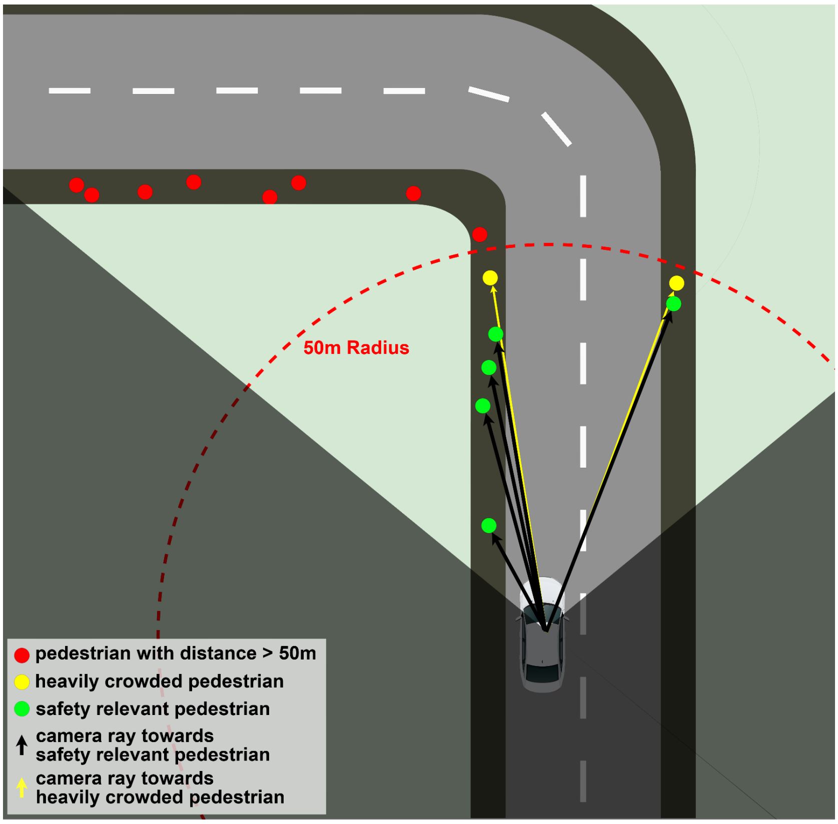

50m Radius
pedestrian with distance > 50m
heavily crowded pedestrian
safety relevant pedestrian
camera ray towards safety relevant pedestrian
camera ray towards heavily crowded pedestrian

Figure 4.5: A bird's-eye view of the driving scenario depicted in Figure 4.4, emphasizing the 50-meter radius and the viewing rays of the camera angle.

4.1.2 Detection Performance Evaluation

The main goal of a detector, as defined by PDSM, is to detect every safety-relevant pedestrian with an IoU value of at least 0.25 while having no FP detections. PDSM is utilizing a lower IoU threshold compared to other object detection metrics, which generally threshold at 0.5 [7, 8, 6]. The reasoning behind this is simply that because many pedestrian instances appear occluded, the detectors may struggle to match the predicted full-body bounding box with the ground truth annotation. This often leads to detections being matched at a lower IoU value, marking the detections as FPs and the corresponding ground truths as FNs. Furthermore, for the scope of this study, the task of detecting pedestrians within an image is preferred over the task of precise localization, making the lower IoU threshold justified.

PDSM defines the following evaluation outcomes:

- True Positive (TP) is a detection that matches a pedestrian ground truth bounding box, regardless of whether it is safety-relevant or not, with an IoU of at least 0.25.

- Safety Relevant True Positive (SRTP) is the same as True Positive (TP) with the additional condition that the pedestrian has to be safety-relevant as of PDSM. All Safety Relevant True Positives (SRTPs) are automatically True Positives (TPs); however, this is not always the case the other way around.

- False Positive (FP) is a detection that does not match any ground truth bounding box at all, or the resulting IoU is lower than 0.25.

- False Negative (FN) is a ground truth bounding box that belongs to a safety-relevant pedestrian and has not been matched with any detection.

PDSM deviates even further from other standard metrics, which typically describe the detection performance as a single value across several classes, confidence thresholds, and IoU thresholds [7, 8, 6]. While these metrics can be quite useful for comparing different sorts of DNN architectures and their detection performance on general object detection tasks, they should not be used when it comes to the safety evaluation of a model that will be deployed in a safety-critical domain. Contrary to this, PDSM evaluates each model in its final deployment state, which means that there is a single confidence threshold that filters out the final detections made by the model and a single IoU threshold for matching detections with ground truth annotations. Setting such strict rules enables PDSM to reduce the noise and give a clear estimation of the detection performance for a given model. Furthermore, it aids in the identification of PLFs. Since each pedestrian instance has several factors linked to it (e.g. occlusion ratio), it is straightforward to keep track of the safety-relevant pedestrians that the model was unable to detect (FNs) and analyze their factors. This approach can be further extended towards tracking different value ranges for a single factor and the corresponding evaluation data-points, i.e., the evaluation outcome (TP, SRTP, FP or FN) for a pedestrian instance that falls into the given value range for a given factor. A subsequent analysis of the detection performance at different values of a factor could reveal its influence on the model's detection performance.

PDSM defines the final detection performance by these three metrics:

- $Precision = \frac{TPs}{(TPs+FPs)}$
- $Recall = \frac{SRTPs}{(SRTPs+FNs)}$
- $F1\text{-}Score = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision+Recall)}$

Note how the precision takes all TPs into account, regardless of whether they are safety-relevant or not. However, when it comes to the recall, only SRTPs are taken into account since PDSM focuses on the detection of all safety-relevant pedestrians. The F1-Score, which combines precision and recall into a single metric by computing their harmonic mean, serves as the primary metric for PDSM.

4.2 Performance Limiting Factors (PLFs)

There is no clear definition of what a Performance Limiting Factor (PLF) is. For the scope of this study a PLF can be loosely described as a factor whose properties impact the detection performance. To analyze the impact of such a factor, one would intuitively study the correlation between various value levels of that factor and the corresponding detection performance. However, one has to consider that the distribution of factor values within the training dataset might be skewed, resulting in very poor data coverage for certain value levels and thus poor detection performance. In such an event, it would be wrong to identify the factor as a PLF, since the main source of the poor performance comes from the lack of training data and not the factor properties themselves [52]. Furthermore, there is also a risk of overfitting to frequent factor values, which is mostly caused by a lack of diversity and a poor distribution of factor values in the training data [52]. Therefore, a more precise definition for PLFs can be derived.

Definition 4.3. This study defines a PLF as a factor that, at certain parameter values, evidently leads to a drop in the detection performance, which cannot be directly linked to poor data coverage within the train dataset [52].

While this definition still lacks coherence, it should be regarded as a first step towards forming a generally accepted definition and serve as argumentative guidance for this work.

This study distinguishes between two types of factors. Object-based factors, like, for example, the bounding box aspect ratio, are extracted per object and therefore linked to individual pedestrian instances. Scene-based factors, on the other hand, are extracted on behalf of the whole input image. A good example of a scene-based factor is the intensity of contrast present in the image. The factors can be further grouped based on the properties they address. They include pixel intensity properties, geometrical properties, and meta annotations. The following sections will cover the 21 factors that are being analyzed in this study.

4.2.1 Pixel Intensity Properties

These types of factors are extracted based on the pixel intensity values of the image and therefore rely on image processing techniques.

Edge Strength

As its name suggests, the edge strength is a factor that quantifies the edge frequency within an image, thus being a scene-based factor. The first step in computing the edge strength of an image is to convert the image I_{RGB} to grayscale producing I_G. Using the Sobel filter [53], one can extract the horizontal and vertical edges $d_x(I_G)$ and $d_y(I_G)$ that are present in an image:

$$d_x(I_G) = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} * I_G, \tag{4.1}$$

$$d_y(I_G) = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} * I_G. \tag{4.2}$$

To combine the two resulting images $d_x(I_G)$ and $d_y(I_G)$, one has to compute the magnitude [54] using the following formula:

$$M = \sqrt{d_x(I_G)^2 + d_y(I_G)^2}, \tag{4.3}$$

where M represents a vector of size $h \cdot w$, where h is the height of the input image I_G and w is the width. The values of this vector are in the range between 0 and 255, indicating the edge strength at each pixel. The final step for computing the edge strength $e(I_G)$ is to convert the magnitude vector into a single value by computing the mean and then normalizing, using the following formula:

$$e(I_G) = \frac{1}{A(I_G) \cdot 255} \sum_{i=1}^{A(I_G)} M_i, \tag{4.4}$$

where $A(I_G)$ represents the pixel area of the input image ($A(I_G) = h \cdot w$) and M_i represents the pixel value at the i-th position of the magnitude vector M from equation (4.3).

Boundary Edge Strength [†]

The boundary edge strength is an object-based factor that quantifies the edge strength of the boundary separating the pedestrian from its background. The first step in computing the boundary edge strength is to compute the magnitude vector M for the entire image I_G using the formula (4.3). This magnitude vector must be further filtered to contain only pixel values that are located on the boundary of a specific pedestrian instance, whose boundary edge strength is to be computed. In order to obtain such a boundary mask, one has to apply the dilation and erosion operators to the binary pedestrian instance segmentation mask S respectively, by using the following formulas:

$$S_{dilated} = S \oplus \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \; S_{eroded} = S \ominus \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \tag{4.5}$$

where $\oplus$ denotes the dilation operation and $\ominus$ denotes the erosion operation [55]. By subtracting the eroded version of the binary instance segmentation mask from the dilated version, one can obtain the boundary mask:

$$S_{boundary} = S_{dilated} - S_{eroded}. \tag{4.6}$$

The process of computing a boundary mask is visualized in Figure 4.6. Next, the boundary mask $S_{boundary}$ is used to remove all values from the magnitude vector M, that do not spatially belong to the pedestrian boundary. Such a filtering function can be described as:

$$F_S(M) = \{M_i \in M \mid i \in [1, ..., h \cdot w], \; S_i = 1\}, \tag{4.7}$$

where S denotes the binary segmentation mask for filtering and M the magnitude vector, which is being filtered. The last step is then to just apply formula (4.4) on the filtered magnitude vector $F_{S_{boundary}}(M)$, in order to obtain the final boundary edge strength for a pedestrian instance. Figure 4.7 visualizes the magnitude vector M for a pedestrian bounding box, before and after applying the boundary mask from Figure 4.6.

Background Edge Strength [†]

This factor is highly similar to the boundary edge strength, and as such, it is an object-based factor. In fact, the only difference between these two factors

[†] Novel Factor.

is that instead of filtering the magnitude vector M with the boundary mask $S_{boundary}$, one must use the inverted version of the dilated instance segmentation mask $S_{dilated}$ from equation (4.5). This produces a magnitude vector of the background within a pedestrian's bounding box, as shown in Figure 4.7. Just as before, one can apply formula (4.4) on the filtered magnitude vector $F_{S^{-1}_{dilated}}(M)$ to compute the final background edge strength.

Contrast

This scene-based factor is responsible for tracking the contrast of a given image. The contrast can be simply described as the degree to how evenly distributed the pixel intensity values within an image are [56]. In statistics, this is basically the standard deviation of pixel intensity values from a given grayscale image I_G. The resulting standard deviation can be normalized through division by a constant factor. This constant factor is equivalent to the maximum standard deviation, which for a standard pixel range (0 to 255) corresponds to:

$$73.9 = \sigma(\{0, ..., 255\}]). \tag{4.8}$$

The following formula computes the contrast factor for a given grayscale image:

$$c = \frac{1}{73.9}\sigma(I_G). \tag{4.9}$$

Contrast to Background [†]

Contrast to background is an object-based factor that quantifies the difference in contrast between foreground and background of a pedestrian bounding box. The first computation step is to convert the bounding box image to grayscale. To separate the foreground from the background, one can use the filtering function described in (4.7). The only difference is that instead of using the magnitude vector M, one simply uses the grayscale bounding box image. The binary instance segmentation mask can be used to filter the foreground, and by inverting the mask, one can filter the background. The next step is to compute the contrast for the filtered foreground and background images using the formula (4.9). To obtain the final value of the contrast to background c_{dif}, one needs to compute the absolute difference between foreground and background contrast:

$$c_{dif} = |c_F - c_B|, \tag{4.10}$$

where c_F is the foreground contrast and c_B is the background contrast. Figure 4.8 visualizes the grayscale foreground and background for a given pedestrian bounding box image.

[†] Novel Factor.

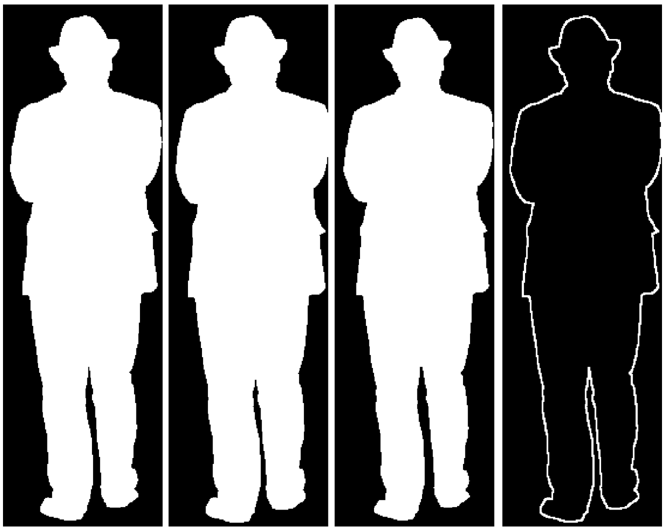


Figure 4.6: Looking from left to right, the first image visualizes the binary instance segmentation mask of a pedestrian bounding box. The second and third images visualize the dilated and eroded versions of the binary instance segmentation mask. The fourth image depicts the boundary mask, which is the result of subtracting the eroded version from the dilated version of the binary instance segmentation mask.

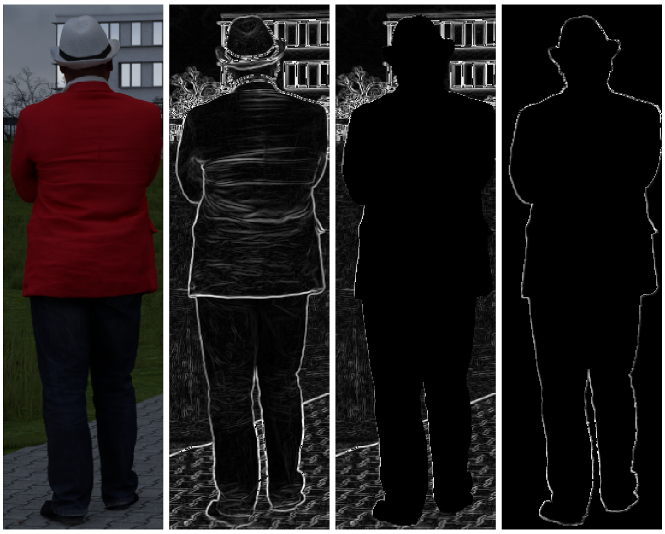


Figure 4.7: Looking from left to right, the first image illustrates a pedestrian bounding box. The second image visualizes the corresponding magnitude vector M. The third image shows the magnitude vector M after it has been filtered by the dilated instance segmentation mask, resulting in the magnitude vector of the background. Applying the boundary mask to the initial magnitude vector yields the fourth and final image.

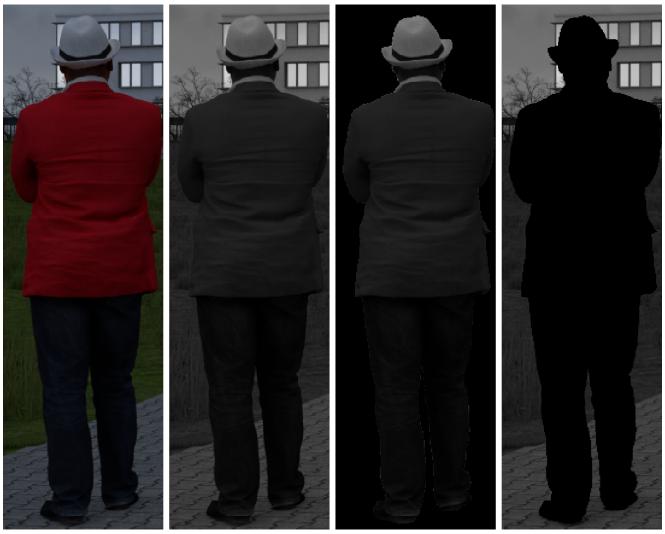


Figure 4.8: Looking from left to right, the first image features a pedestrian bounding box. The second image features a grayscale version of the first image. The third and fourth images visualize the foreground and background of the grayscale pedestrian bounding box.

Brightness

The brightness factor is responsible for tracking the overall pixel brightness of a given image and is therefore a scene-based factor. In statistical terms, it can be simply described as the mean of the pixel intensity values from a given grayscale image I_G [56]. Dividing the mean by a constant factor of 255 (the maximum brightness for the standard pixel range) yields the normalized brightness value:

$$b = \frac{1}{255}\mu(I_G). \tag{4.11}$$

Object Entropy

This factor was inspired by the recent work of Pliusshch *et al.* [49]. It is an object-based factor responsible for tracking the Shannon entropy of a pedestrian bounding box image. Entropy, which measures statistical randomness, can be used to describe the texture of the input image [57]. It can be computed using the following formula:

$$H = -\sum_{k=0}^{255} P_k \cdot \log_2(P_k), \tag{4.12}$$

where P_k refers to the frequency of a pixel value $k \in \{0, ..., 255\}$ within a given grayscale image [58].

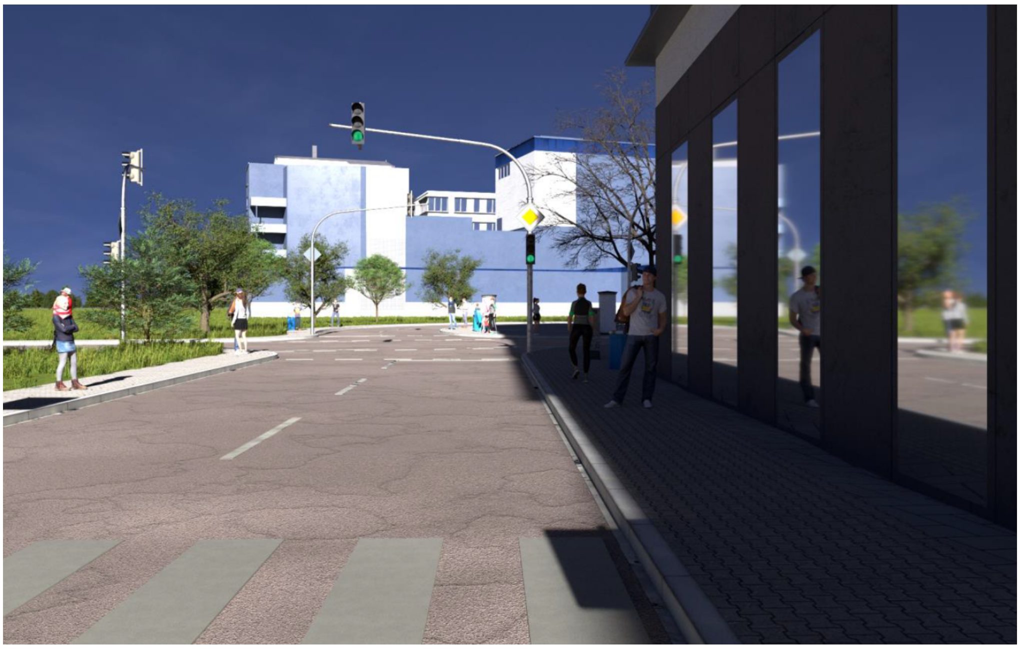


Figure 4.9: Sample image taken from the KI-A dataset [5], showing the difference between pedestrians with high brightness (staying in the sun) and pedestrians with low brightness (staying in the shadow).

Foreground Brightness

This object-based factor is used for tracking the overall pixel brightness of a pedestrian's bounding box foreground. It is computed in analogy to the general brightness formula (4.11) with the exception that the input image I_G is filtered to contain only the foreground of a pedestrian bounding box, using the binary instance segmentation mask. This filtering process can be described as $F_S(I_G)$. Figure 4.9 visualizes several pedestrians with varying foreground brightness values.

4.2.2 Geometrical Properties

Factors belonging to this group are directly linked to the geometrical properties of the object's bounding box and its location with respect to the 2D image.

Bounding Box Height

As its name already suggests, this factor is responsible for tracking the pixel height of a pedestrian bounding box and is therefore an object-based factor.

Crowdedness [†]

The crowdedness is a novel object-based factor that estimates the degree to which a pedestrian appears crowded within a given image. Unlike other factors, whose values are normalized in the range from 0 to 1, crowdedness is an incremental factor, which means that each additional pedestrian instance might increase the total crowdedness of another pedestrian instance. For a pedestrian to be considered crowded, its bounding box has to overlap with another pedestrian's bounding box. Note that the term "crowded" is not to be confused with the term "heavily crowded" from PDSM. As already discussed in section 4.1.1, the overlap of a pedestrian's bounding box with another bounding box is equivalent to the pixel area of the intersection divided by the total pixel area of the bounding box at hand. A direct consequence of this is that the overlap is computed for each bounding box separately, which means that two overlapping bounding boxes each have two distinct overlap values:

$$O_i = \frac{A(BB_i \cap BB_j)}{A(BB_i)},\ O_j = \frac{A(BB_i \cap BB_j)}{A(BB_j)}, \tag{4.13}$$

where BB_i and BB_j denote two pedestrian bounding boxes and A denotes the pixel area of the given bounding box image. The more bounding box area is covered by another bounding box, the higher the overlap value will be, with the maximum overlap value being 1, meaning the bounding box is fully covered. While the overlap measure already gives a good indication of the overall crowdedness, it does not account for the case where the pedestrians are located far from one another, resulting in large size differences between the bounding boxes. In such scenarios, the overlap value will often be high as the smaller bounding box is easily covered, resulting in higher crowdedness scores. Because of this, each overlap value is weighted by the respective size ratio between the two bounding boxes, using the following formula:

$$R = \frac{min(A(BB_i), A(BB_j))}{max(A(BB_i), A(BB_j))}. \tag{4.14}$$

The size ratio will be a number between 0 and 1, where a value of 1 indicates that the two bounding boxes are of equal pixel size and therefore highly likely to be not too far away from one another. The final crowdedness score of a pedestrian instance can be computed using the following formula:

$$C_i = \sum_{j \in \{1..N\}/\{i\}} \frac{A(BB_i \cap BB_j)}{A(BB_i)} \cdot R, \tag{4.15}$$

where C_i is the crowdedness score for a pedestrian instance indexed with i, N is the total amount of pedestrian instances within the image and R represents the ratio from the formula (4.14). An example of a detector output for a scenario featuring pedestrian crowds is given in Figure 4.10.

[†] Novel Factor

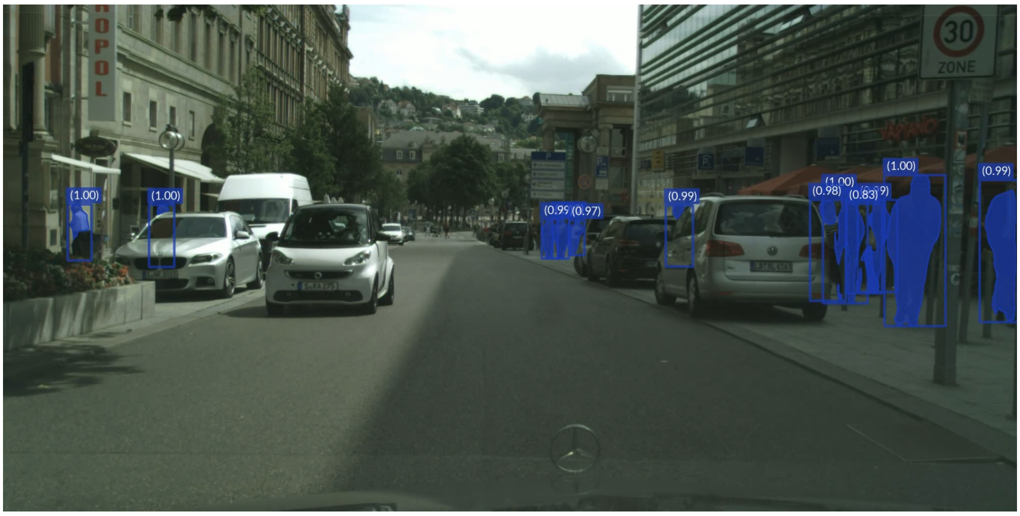


Figure 4.10: Visualization of pedestrian detections, including instance segmentations made by a model on an image sample from the CityPersons dataset [3], featuring several pedestrian instances with varying crowdedness scores, occlusion ratios, and truncation values.

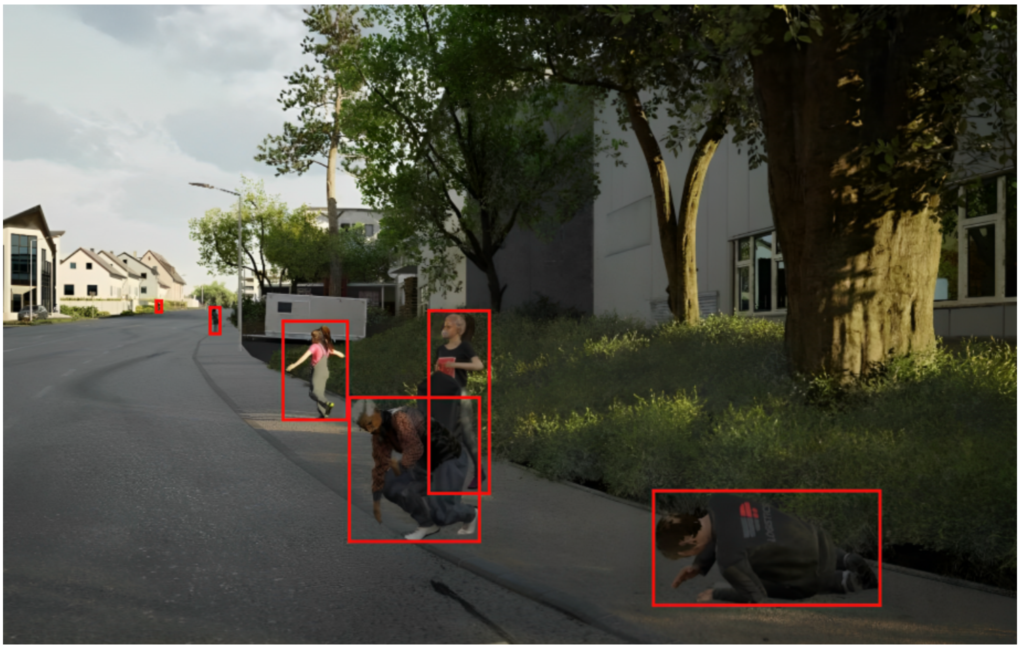


Figure 4.11: Sample image taken from the KI-A dataset [5], featuring several pedestrian bounding boxes with varying distances, bounding box heights, bounding box aspect ratios and the amount of visible instance pixels.

Occlusion Ratio

A particularly difficult task for an object detection model is to predict the full-body bounding box for an occluded pedestrian instance. The occlusion ratio is an object-based factor that measures the ratio between the amount of visible instance pixels and the total amount of instance pixels without any occlusions. Its value can be interpreted as the degree to which a pedestrian instance appears occluded, with the value of 0 representing an unoccluded pedestrian and the value of 1 representing a fully occluded one.

Truncation

Truncation is a binary object-based factor whose purpose is to indicate whether the pedestrian's bounding box has been truncated as a consequence of being located outside the image's borders. To determine the truncation value (0 or 1) for a pedestrian instance, one needs to check whether portions of the bounding box are located outside the image's borders, suggesting that the corresponding pedestrian instance is probably appearing truncated. An example for a truncated pedestrian instance is given in Figure 4.10 as the rightmost bounding box.

Bounding Box Aspect Ratio

Pedestrians are highly likely to appear in various sizes, forms, and poses, resulting in varying bounding box aspect ratios, which a DNN is required to handle. Accordingly, the bounding box aspect ratio is another object-based factor that is being tracked. Its value can be simply computed by dividing the bounding box's pixel width by its pixel height. Several examples of varying bounding box aspect ratios are given in Figure 4.11.

Visible Instance Pixels

Another factor that is highly impacted by the varying appearances of a pedestrian instance within an image, is the object-based factor responsible for tracking the amount of visible instance pixels. Its value is equivalent to the amount of positive pixel values within the binary instance segmentation mask.

Distance

Distance is an object-based factor responsible for tracking the distance between a particular pedestrian instance and the camera. Its value is described in meters.

4.2.3 Meta Annotations

The final group of factors addresses common environmental properties, which cannot be directly computed or extracted from the images themselves. Such scene-based factors are within the scope of this study referred to as "meta annotations". Note that all meta annotations are extracted from the synthetic KI-A dataset [5], which will be later introduced in section 5.1.1, and are therefore subjected to predefined value ranges. Furthermore, their values are automatically assigned by the KI-A simulation for each image sample.

Lens Flare Intensity

Depending on their size and position, lens flare artifacts may have a negative impact on the detection performance of a model and are therefore a potential PLF. As this scene-based factor belongs to the group of meta annotations, its value is tracked by the KI-A simulation for each individual image. The value range is between 0 and 1. Figure 4.12 is an exemplary image that contains lens flare artifacts.

Vignette Intensity

Vignetting directly affects the pixel area around the image's borders, making them appear darker. This effect might lead to unwanted consequences for a detection model and is therefore being analyzed as another meta annotations factor. The vignette intensity is described as a value ranging from 0 to 1.

Fog Intensity

Fog is one of the most notorious weather conditions, known for reducing overall visibility and the appearance of images by adding a thick white layer. The question at hand is: how well do current detectors adapt to detecting pedestrians in foggy conditions? The fog intensity factor, ranging from 0 to 1, is responsible for quantifying the amount of fog that is added within the KI-A simulation and is hence another scene-based factor. An example of a moderately foggy scene is given in Figure 4.13.

Sky Type

Sky type describes the appearance of the sky in the form of three possible values: "clear", "partially clouded", and "fully clouded".

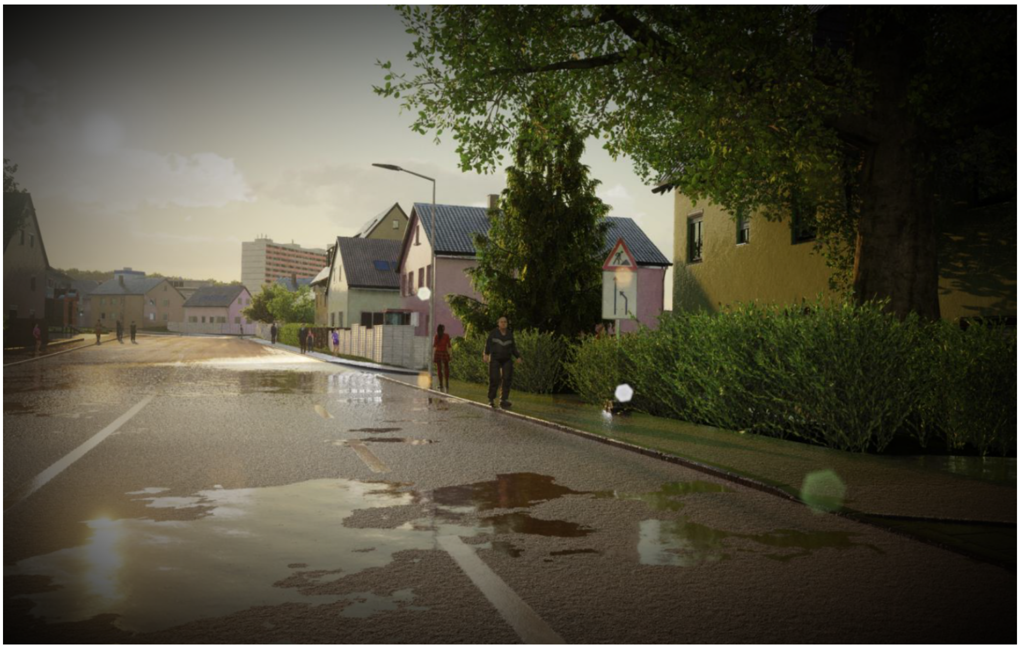


Figure 4.12: Image sample taken from the KI-A dataset [5], which contains lens flare at a medium sun position causing a vignetting effect.

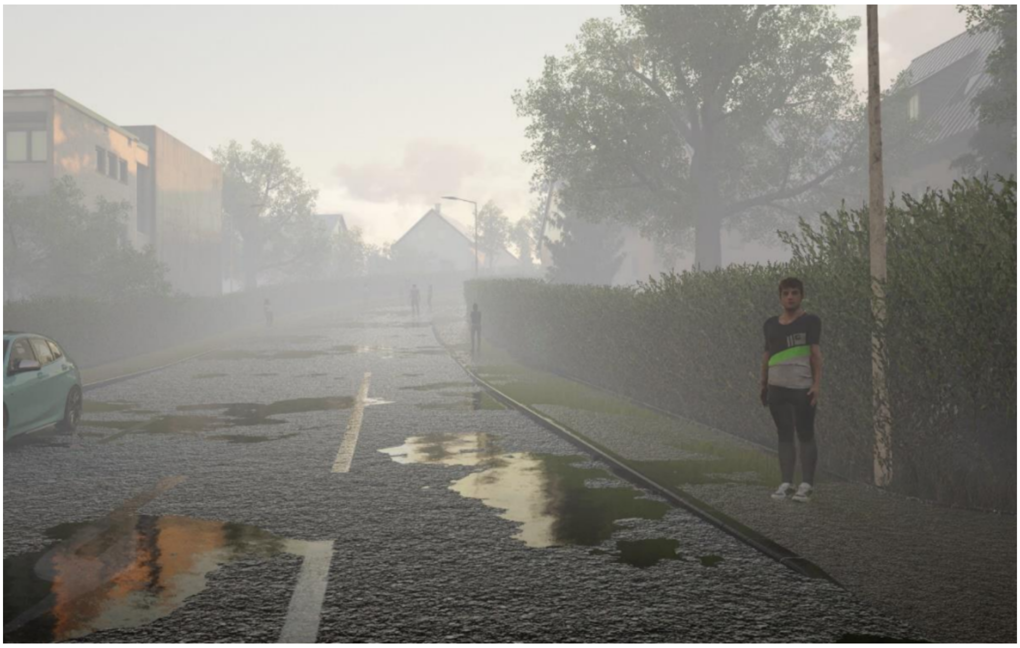


Figure 4.13: Image sample taken from the KI-A dataset [5], illustrating moderate foggy conditions with wet roads and puddles.

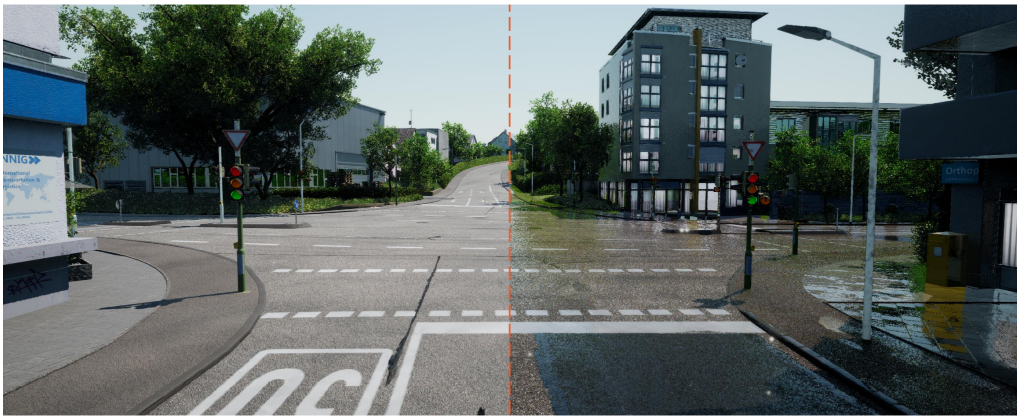


Figure 4.14: Visualization of a road crossing taken from the KI-A dataset [5], depicting dry road conditions (left side) and slightly moist road conditions (right side).

Daytime Type

The daytime is a categorical factor, since it can only have the following three possible values: "day", "medium sun position", and "low sun position". In this case, the medium sun position describes the beginning of sunset or the end of sunrise, adding a soft orange tone to the image. Low sun position, on the other hand, describes the end of sunset or the start of sunrise, with the overall image being darker with a stronger orange tone.

Wetness Type

Wetness type is yet another categorical meta annotation that describes the degree to which the ground appears wet in an image. The three possible categories are: "dry", "slightly moist", and "wet with puddles". Figure 4.14 shows an example for dry and slightly moist roads, while Figures 4.12 and 4.13 show examples of roads that are wet with puddles.

Chapter 5

Experiment Setup

5.1 Datasets

In order to properly assess the influence of the factors from section 4.2 on the detection performance, one needs to analyze the DNN's behavior in various driving scenarios. For this purpose, two pedestrian detection datasets are used for the experiments, the first one being synthetic and the second being real. Both of them are situated within urban road scenarios and offer a very high variety of unique driving situations. Furthermore, the use of a synthetic and a real-world dataset adds additional value to this study by allowing for a direct comparison with respect to DNN behavior.

5.1.1 KI Absicherung (KI-A)

The KI Absicherung (KI-A) dataset [5] is a large-scale synthetic pedestrian detection dataset introduced by the KI Absicherung project [59], which is funded by the German Federal Ministry for Economic Affairs and Climate Action. The primary goal of this dataset is to provide high variability and control over environmental factors with highly detailed annotations to allow for safety-oriented training and validation of the AI functions [60]. Some of the important dataset aspects include, but are not limited to, weather diversity, daytime diversity, camera sensor variations, ground wetness, variations in road appearance, and many more. The pedestrian instances that appear within the image samples also show a high degree of diversity, featuring various poses that are usually difficult to capture in real-world datasets like the ones shown in Figure 5.1. Overall, the KI-A dataset provides a very strong foundation for conducting a PLF analysis due to its large scale and highly detailed meta-data that is provided for each sample image. For this study, only the "human" class was used, which includes every human instance within an image sample. The dataset supports, among others, the three detection tasks that are studied in this work, including 2D Object Detection (2D-OD), 2D Instance Segmentation (2D-IS), and 2D Keypoint Detection (2D-KD). Furthermore, as of the time this study was conducted, the

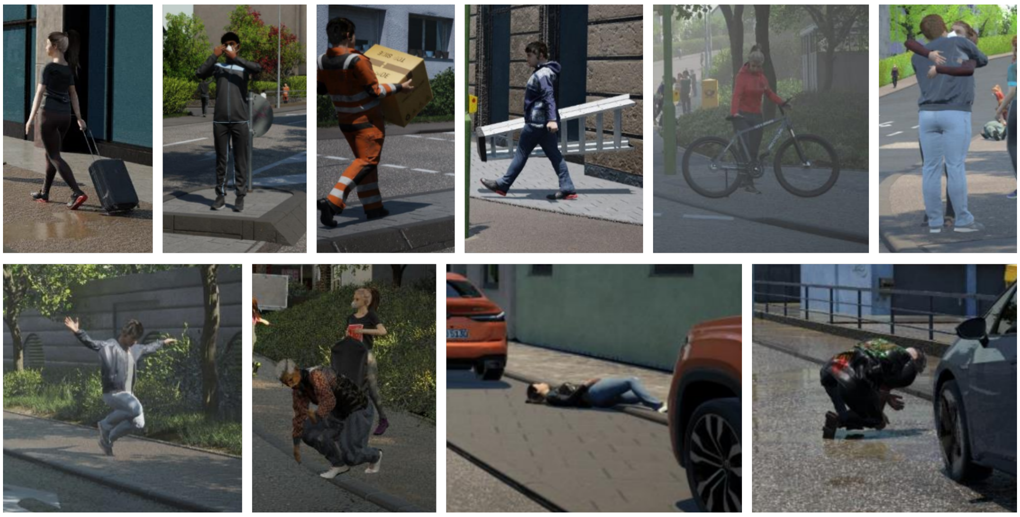


Figure 5.1: Several examples of pedestrian instances from the KI-A dataset showcasing high variations in their appearance.

KI-A dataset was actively being developed, continuously increasing its size. The Table 5.1 offers an overview of the amount of train, validation, and test samples within the KI-A dataset at the time of conducting the experiments.

Train Samples	Validation Samples	Test Samples
56,161	15,106	145,518

Table 5.1: Amount of train, validation, and test samples from KI-A, that were used for the experiments in this work.

5.1.2 CityPersons

CityPersons has been created upon the CityScapes dataset [26] focusing specifically on the task of pedestrian detection. The image samples were recorded over several months (spring, summer, and fall) in urban street scenes from 50 different German cities at daytime [3]. Unfortunately, there are no meta annotations within CityPersons making it impossible to track any of the environmental factors that were introduced in section 4.2.3. The dataset supports two detection tasks, including 2D-OD and 2D-IS. It contains the following six classes: "ignore regions", "pedestrians", "riders", "sitting persons", "other persons with unusual postures", and "group of people". For the scope of this study, the class "ignore regions" has not been considered, and the remaining classes have been grouped into a single "human" class in analogy to the KI-A dataset. However, since the class "group of people" does not represent a single pedestrian instance but rather a group of pedestrians, the evaluation process has been adjusted accordingly. This just means that the ground truth annotations that are marked as "group of people" are allowed to be matched multiple times instead of just once to account for multiple pedestrian instances within a single bounding box annotation. It should be further mentioned that a very small portion of pedestrian instances falls into this category and that all of them are situated in the far distance, thus being annotated as a "group of people" for convenience. CityPersons contains a total of 5,000 image samples with fine annotations. However, 1,500 samples from the test set do not have publicly available annotations since the authors offer an official evaluation server for submissions. Therefore, for this study, the official validation split acted as the test split, while 300 randomly sampled images from the train split acted as the new validation split. Table 5.2 provides a summary of the number of CityPersons samples within the final train, validation, and test splits.

Train Samples	Validation Samples	Test Samples
2,700	300	500

Table 5.2: Amount of train, validation, and test samples from CityPersons, that were used for the experiments in this work.

A major issue with CityPersons was that the pedestrian annotations did not include any occlusion information except an approximation of the full-body bounding box. The solution to this problem is another major contribution of this work, in the form of a new occlusion estimation regression model. The basic idea is to approximate the occlusion ratio for a given pedestrian instance based on the following features:

- **Pixel area** of the full-body bounding box.
- **Amount of visible instance pixels** within the pedestrian bounding box.
- **Empty rows ratio**, which is simply the amount of bounding box pixel rows that do not contain the pedestrian instance (can be extracted based on the binary instance segmentation mask), divided by the total amount of bounding box pixel rows.
- **Empty columns ratio**, which is similar to **empty rows ratio**, but focuses on columns rather than rows.

In order to be able to estimate the occlusion ratio, a pedestrian instance is required to have a full-body bounding box annotation and a binary instance segmentation mask, which is the case for all pedestrian instances from CityPersons. To train the linear regression model, a dataset was collected featuring examples of annotated pedestrian instances from the KI-A dataset. In addition to the

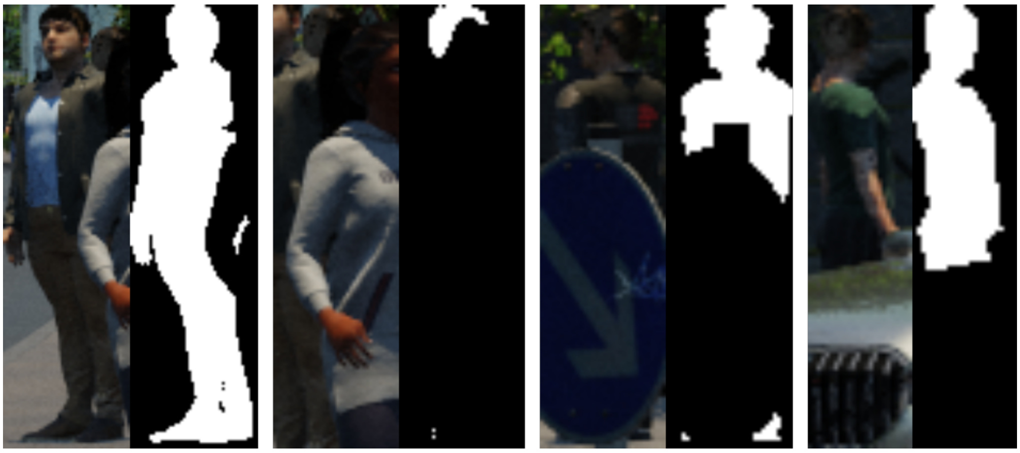


Figure 5.2: Several examples of pedestrian bounding boxes and their binary instance segmentation masks taken from KI-A. Looking from left to right, the occlusion ratios of the four pedestrian instances have been approximated by the occlusion estimation model to be 0.12, 0.86, 0.5, and 0.43, respectively. The ground truth occlusion ratios for these instances are 0.22, 0.95, 0.68, and 0.38, respectively.

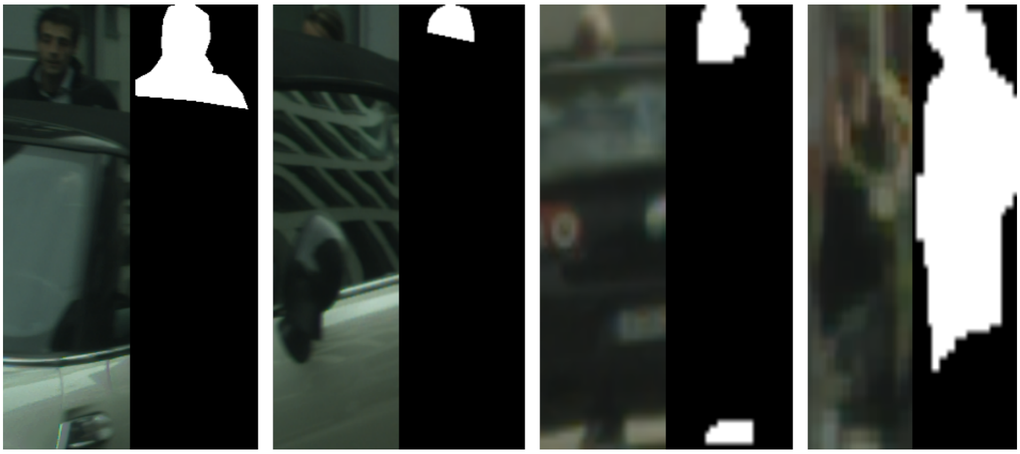


Figure 5.3: Several examples of pedestrian bounding boxes and their binary instance segmentation masks taken from CityPersons. Looking from left to right, the occlusion ratios of the four pedestrian instances have been approximated by the occlusion estimation regression model to be 0.84, 0.99, 0.82, and 0.25, respectively.

aforementioned features, contains the KI-A dataset also the corresponding labels in the form of the occlusion ratio, which is computed by the KI-A simulation for each pedestrian instance. A total of 4,160,746 such pedestrian samples have been collected, and the resulting dataset has been split into train and test samples as shown in Table 5.3.

Train Samples	Test Samples
3,320,676	830 170

Table 5.3: Amount of train and test samples of the newly collected dataset, that was used to train the occlusion estimation regression model.

The occlusion estimation model has been trained using the Scikit-learn library [61]. The final regression coefficients correspond to the following values:

- **Pixel area coefficient:** $351 \cdot 10^{-8}$.
- **Amount of visible instance pixels coefficient:** $-908 \cdot 10^{-8}$.
- **Empty rows ratio coefficient:** 0.719.
- **Empty columns ratio coefficient:** 0.199.
- **Intercept:** 0.114.

A subsequent performance evaluation of the occlusion estimation model on the test set yielded a Mean Absolute Error (MAE) of 0.096, a Mean Squared Error (MSE) of 0.017, and a Root Mean Squared Error (RMSE) of 0.132. This means that the occlusion ratios predicted by the occlusion estimation model are off by ±9.6% on average, which is a very acceptable performance. However, since the evaluation has solely been conducted on the dataset that was also used for training, there is still uncertainty about the generalizability of this model towards other datasets. Finally, the occlusion estimation model has been applied to the CityPersons dataset to generate approximations of the occlusion ratio for each pedestrian instance. Figures 5.2 and 5.3 show several examples of pedestrian instances from KI-A and CityPersons, as well as the estimated occlusion ratios obtained by the occlusion estimation model.

5.2 Pedestrian Detectors

This work studies the behavior of six commonly used DNN architectures. Each of them has been selected to cover a wide range of different state-of-the-art approaches for the general task of object detection. This includes one-stage and two-stage detectors, as well as anchor-based and anchor-free approaches. The following one-stage object detection architectures are used for the experiments: SSD300 [31], RetinaNet [62], and FCOS [63]. This list of pedestrian detectors is further extended by the following two-stage architectures: Faster R-CNN [29], Mask R-CNN [64], and Keypoint R-CNN [64]. All of these models are anchor-based except for FCOS, which is an anchor-free approach. Moreover, all of the aforementioned architectures have the same backbone model in the form of a ResNet50 architecture [65], which has been previously pretrained on the ImageNet dataset [66]. The pretrained model weights and model implementations have been obtained by the Torchvision package [67] from the PyTorch framework [68]. Furthermore, there are three detection tasks studied in this work, including 2D-OD (supported by all architectures), 2D-IS (supported by Mask R-CNN), and 2D-KD (supported by Keypoint R-CNN). The two-stage detectors used in this study all come from the R-CNN family [69] and thus have very similar architectures. In fact, Mask R-CNN and Keypoint R-CNN are just extensions of the Faster R-CNN architecture that support 2D-IS and 2D-KD, in addition to the main 2D-OD task. This offers another interesting aspect of this study by allowing for a direct comparison between these highly similar DNN models on their robustness towards PLFs. As 2D-IS and 2D-KD add further complexity to the detectors by optimizing them towards the identification of further pedestrian semantics, it can be assumed that the Mask R-CNN and Keypoint R-CNN architectures have a clear advantage in achieving superior detection performance; however, it is unclear whether such an effect can also be observed with respect to PLF robustness.

5.3 Detector Training

This section first introduces the training approach for F1-Score maximization, which can also support the maximization of any other evaluation metric. The goal of this training approach is to ensure that the DNN's behavior is analyzed at its highest level of performance with respect to PDSM. The second part of this section introduces the actual hyperparameter setup used for the final trainings of the DNNs from section 5.2.

5.3.1 Training Approach for F1-Score Maximization

The training approach for F1-Score maximization requires a standard dataset split consisting of train, validation, and test data. In the first step, each detection model is trained on several sets of hyperparameters over several epochs on the train split. Each epoch ends with the model weights being stored and evaluated on the validation split, with respect to PDSM. As models also assign a confidence score to each detection, it is important to determine a model's individual confidence threshold at which the detection performance is maximized.

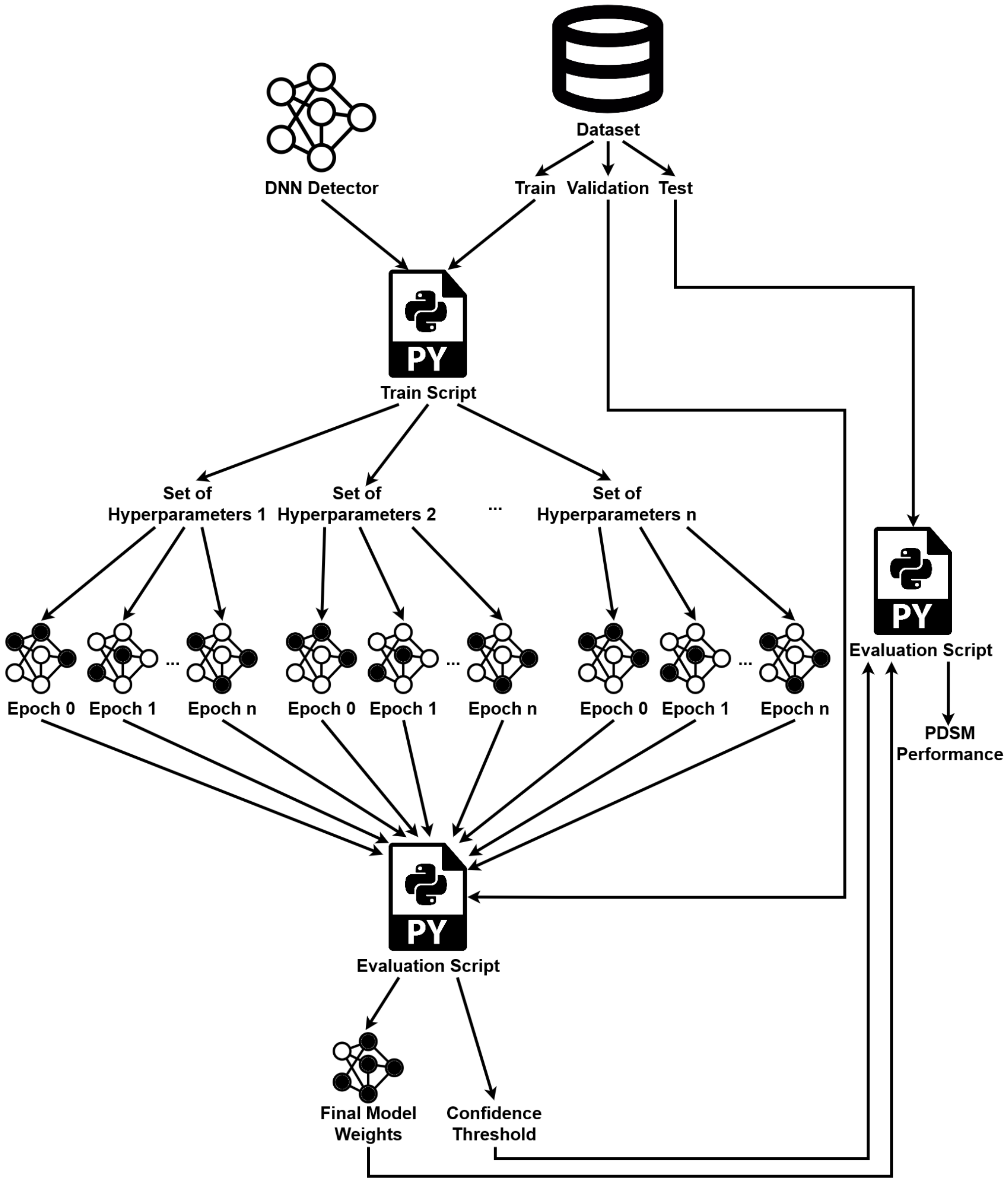


Figure 5.4: Visualization of the training approach for F1-Score maximization with respect to PDSM.

For this purpose, each evaluation is subjected to several confidence thresholds ranging from 0 to 1.0 with a step size of 0.05. Once all trainings are finished, the next step is to determine which model weights were able to achieve the single highest F1-Score on the validation split and at what confidence threshold they were able to do so. The last step is then to just evaluate these final model weights on the test split using the predetermined confidence threshold to compute the final detection performance with respect to PDSM. Figure 5.4 offers a visual explanation of the training approach for F1-Score maximization.

5.3.2 Training and Hyperparameter Setup

The final training setup involved training the models on the large-scale synthetic KI-A dataset first, then applying transfer learning to fine-tune the model weights on CityPersons due to significant size disparities between the two datasets. The only exception to this was the Keypoint R-CNN model, which requires pedestrian keypoint labels during the training process, thus being trained only on KI-A. All trainings were conducted using the PyTorch framework [68]. During the training process, each sample image has been subjected to a random horizontal flip before being inferred into the models in order to increase the total number of image samples on which the models have been optimized. The process of hyperparameter tuning had to be reduced to just two sets of hyperparameters because of limitations in both time and computational power. The first set contained all visible pedestrians that are annotated, while the second hyperparameter setup restricted the pedestrian instances to have at least 50 instance pixels, a maximum occlusion ratio of 0.95, and be located at most 75 meters away from the camera. The motivation for such a filtering of pedestrian instances during the training stems from the fact that the synthetic KI-A dataset features numerous pedestrian annotations that are located in the far distance (non-safety-relevant as of PDSM) making it difficult to detect all instances even for a human, and therefore it cannot be ruled out that their presence might confuse the detectors, leading to lower detection performance. Figure 4.4 from Chapter 4 visualizes some of the aforementioned pedestrian instances that are barely visible in the form of the six leftmost bounding boxes. The hyperparameter values for filtering pedestrian instances were chosen on a best effort basis after manual inspections to be conformant with PDSM and to decrease potential sources of noise, with the end goal of achieving the highest possible PDSM detection performance.

The final hyperparameter setup included the following parameters and values:

- **Minimum object pixels** defines the threshold for filtering pedestrian instances with a too low amount of visible instance pixels. This hyperparameter was tuned on two values, including **1** (no filtering) and **50**. All pedestrian instances, which had a lower amount of visible instance pixels, were filtered out during the training process.

- **Maximum object occlusion** describes the threshold for filtering pedestrian instances with a too high occlusion ratio. This hyperparameter was also tuned on two values including **0.99** (no filtering) and **0.95**. All pedestrian instances, which had a higher occlusion ratio, were filtered out during the training process.

- **Maximum object distance** is yet another hyperparameter that defines a threshold for filtering pedestrian instances. Its purpose is to filter out pedestrian instances that are too far in the distance. This hyperparameter was tuned between the values **100** meters (very few samples are visible at this distance) and **75** meters.

- **Number of epochs** is used to specify how many times the training process should iterate over the training data. All models were trained for **25** epochs on the KI-A dataset, and the best-performing model weights were finetuned on CityPersons for another **50** epochs.

- **Batch size** describes the number of image samples used to train the model in a single optimization step. Its value was set to **4** for all trainings.

- **Optimizer** is the algorithm that is used for optimizing (training) the model weights with respect to the given task using training data. All trainings in this work have been conducted with **Stochastic Gradient Descent (SGD)**.

- **Learning rate** defines the optimization rate at which the model weights are adjusted at each training step. Its value has been tuned in a quick manner to be as high as possible without crashing the training process. On KI-A a learning rate of **0.01** was used for all models except for RetinaNet and SSD300, which used a learning rate of **0.001**. During the finetuning on CityPersons, the learning rate remained the same except for FCOS, which also trained at a learning rate of **0.001**.

- **Weight decay** serves as a simple regularizer for reducing the risk of overfitting by decaying the model weights towards zero at each optimization step. The value of this hyperparameter was set to **0.0001** for all trainings.

- **Momentum** defines the momentum for the SGD optimization algorithm and was set to **0.9** for all trainings.

- **Learning rate scheduler step size** is used to automatically reduce the learning rate each time after a defined number of epochs. For KI-A the learning rate scheduler step size has been set to **5** epochs, while for CityPersons, its value was set to **10** epochs.

- **Learning rate scheduler gamma** is the second parameter that is used to adjust the learning rate during the training. The value of this hyperparameter was set to **0.1** for all trainings, meaning that each time after the amount of epochs described by the learning rate scheduler step size has been processed, the current learning rate gets decayed by multiplying it with 0.1.

Chapter 6

Results & Discussion

6.1 Training Results

The following section presents and discusses the final evaluation results of the best-performing model weights from the training process described in section 5.3. The detection tasks for which the models have been optimized include 2D Object Detection (2D-OD), 2D Instance Segmentation (2D-IS), and 2D Keypoint Detection (2D-KD). Since this study is about the safety analysis of 2D-OD pedestrian detectors, the Mask R-CNN and Keypoint R-CNN models have also been evaluated based upon their 2D-OD pedestrian detection performance. This approach allows for a direct comparison between all six models on the general 2D-OD task and could also reveal the potential benefits of such hybrid models that support 2D-IS or 2D-KD, on the overall detection performance and their robustness towards PLFs. Tables 6.1 and 6.2 summarize the detection performance of the six pedestrian detectors that are studied in this work, on the KI-A and CityPersons test splits, with respect to PDSM. The tables include the best-performing confidence threshold and F1-Score for each model, as well as the final precision and recall values at an 0.25 IoU threshold. The best-performing models with respect to each metric are highlighted as bold values. Tables 6.3 and 6.4 extend these evaluation results by presenting the detection performance with respect to three standard object detection evaluation metrics, including Average Precision (AP) as defined by Common Objects in Context (COCO) [6], Average Precision (AP) as defined by PASCAL Visual Object Classes (PASCAL VOC) [7], and Log Average Miss-Rate (LAMR) as defined by the Caltech pedestrian benchmark [8]. The tables also include the inference speeds of the respective models. All of the aforementioned metrics utilize an IoU threshold of 0.5 except for AP by COCO, which evaluates the detectors over several IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. It should be noted here that the commonly used mean Average Precision (mAP) metric, as defined by COCO is not applicable in this study, as it evaluates the detection performance over multiple object classes, while this work studies the detection performance on just a single "human" class. Further supporting figures related to the training results are included in Appendix A.

Detectors	Task	Conf. Thr.	$F1^{25}_{PDSM}$	$Precision^{25}_{PDSM}$	$Recall^{25}_{PDSM}$
KeypointRCNN	2D-KD	0.85	0.930	**0.978**	0.888
MaskRCNN	2D-IS	0.80	**0.933**	0.977	**0.894**
FasterRCNN	2D-OD	0.80	0.930	0.977	0.888
FCOS	2D-OD	0.50	0.921	0.973	0.874
RetinaNet	2D-OD	0.40	0.893	0.936	0.854
SSD300	2D-OD	0.25	0.739	0.806	0.681

Table 6.1: Evaluation results of the pedestrian detectors on the KI-A dataset, with respect to PDSM.

Detectors	Task	Conf. Thr.	$F1^{25}_{PDSM}$	$Precision^{25}_{PDSM}$	$Recall^{25}_{PDSM}$
KeypointRCNN	2D-KD	-	-	-	-
MaskRCNN	2D-IS	0.80	**0.860**	0.924	**0.805**
FasterRCNN	2D-OD	0.85	**0.860**	**0.928**	0.802
FCOS	2D-OD	0.50	0.851	0.912	0.798
RetinaNet	2D-OD	0.50	0.831	0.926	0.754
SSD300	2D-OD	0.20	0.638	0.744	0.558

Table 6.2: Evaluation results of the pedestrian detectors on the CityPersons dataset, with respect to PDSM.

Almost all of the models achieved very high detection performances on both datasets, demonstrating the effectiveness of the introduced training approach for F1-Score maximization from section 5.3.1. As can be observed from Tables 6.1 and 6.2 the best performing confidence thresholds differ significantly among the individual detectors, while at the same time they appear to deviate just slightly when comparing their values for a single model across the two datasets. Furthermore, the two-stage detectors studied in this work achieved superior detection performance compared to the one-stage detectors. Interestingly, there is only a marginal performance increase observable for the Mask R-CNN and Keypoint R-CNN detectors compared to the baseline Faster R-CNN, which invalidates the initial hypothesis about the benefits of optimizing the detectors towards 2D-IS or 2D-KD, on the general 2D-OD performance. However, it still remains unclear whether such an effect can also be observed with respect to PLF robustness. Unsurprisingly, there is a lower detection performance on CityPersons compared to the KI-A dataset. Besides the basic fact that these datasets come from different domains, with KI-A being synthetically generated and CityPersons being recorded from real-world scenes, there is also a significant difference in the amount of training data that was provided to optimize the detectors. Moreover, there is a divergence between the best-performing models as measured by PDSM and the other metrics from the literature. As can be observed from the plots presented

Detectors	Task	$AP^{50:.05:95}_{COCO}$	$AP^{50}_{PASCAL\ VOC}$	$LAMR^{50}$	Frames/Sec.
KeypointRCNN	2D-KD	0.544	0.887	0.514	9.990
MaskRCNN	2D-IS	0.582	**0.898**	0.500	4.032
FasterRCNN	2D-OD	0.570	0.895	0.509	14.648
FCOS	2D-OD	**0.611**	0.897	**0.320**	16.161
RetinaNet	2D-OD	0.525	0.861	0.399	15.352
SSD300	2D-OD	0.214	0.536	0.856	**97.076**

Table 6.3: Evaluation results of the pedestrian detectors on the KI-A dataset, with respect to AP by COCO (higher is better) [6], AP by PASCAL VOC (higher is better) [7], and LAMR by the Caltech pedestrian benchmark (lower is better) [8]. The rightmost column contains the inference speed for each model in Frames Per Second (FPS) units.

Detectors	Task	$AP^{50:.05:95}_{COCO}$	$AP^{50}_{PASCAL\ VOC}$	$LAMR^{50}$	Frames/Sec.
KeypointRCNN	2D-KD	-	-	-	-
MaskRCNN	2D-IS	0.520	0.803	0.607	6.171
FasterRCNN	2D-OD	0.508	0.802	0.610	16.639
FCOS	2D-OD	**0.563**	**0.830**	**0.384**	16.676
RetinaNet	2D-OD	0.504	0.799	0.426	17.391
SSD300	2D-OD	0.180	0.440	0.835	**106.758**

Table 6.4: Evaluation results of the pedestrian detectors on the CityPersons dataset, with respect to AP by COCO (higher is better) [6], AP by PASCAL VOC (higher is better) [7], and LAMR by the Caltech pedestrian benchmark (lower is better) [8]. The rightmost column contains the inference speed for each model in Frames Per Second (FPS) units.

in Appendix A, the lower IoU threshold that is employed by PDSM does not contribute to significant performance increases compared to the standard IoU threshold of 0.5 that is used by other metrics (AP by COCO is an exception since it uses several IoU thresholds). This consequently means that the divergence between the best performing models, as measured by PDSM and other standard metrics stems from the fact that the other metrics take several confidence thresholds for each of the respective models into account, while PDSM applies a more practical approach by evaluating the detection performance at just a single confidence threshold, which is carefully chosen to maximize the detection performance. All of the reported detection performances were achieved with the hyperparameter set that restricted the pedestrian instances to have at least 50 instance pixels, a maximum occlusion ratio of 0.95, and be located at most 75 meters away from the camera. The only exception to this is FCOS on KI-A, which performed better when trained on all pedestrian instances.

6.2 Performance Limiting Factor (PLF) Analysis

This section covers the Performance Limiting Factor (PLF) analysis for the six pedestrian detectors that are studied in this work, based on the KI-A and the CityPersons datasets. The goal of this analysis is to reveal which of the studied factors impacts the detection performance regardless of the distribution of factor values within the training data as defined by the PLF definition (4.3) from section 4.2. In order to conduct such an PLF analysis, it is first necessary to quantify the individual PLFs within the studied datasets. For this purpose, each pedestrian ground truth annotation has been expanded by its respective object-based PLF values and the PDSM evaluation outcome (TP, SRTP, FP, or FN) for each of the studied models. The same applies for individual image samples, which have been annotated with respect to the studied scene-based factors and are also linked with the aforementioned pedestrian ground truth annotations that are present within the image. To handle all of this data, this study utilized the FiftyOne tool [70], which comes with a built-in MongoDB [71] database that allows efficient lookups and filtering. Each of the 21 factors studied in this work is qualitatively analyzed based on correlation graphs that visualize the relationship between factor values and detection performance. This approach enables the identification of non-linear correlation patterns, which cannot be described by the correlation coefficient in a meaningful way. Moreover, as per definition, PLFs are expected to correlate with the detection performance, while at the same time the distribution of factor values within the training data must not be the root cause for this effect. Hence, a qualitative analysis of the correlation graphs is a secure way of estimating whether a particular factor fulfills the requirements for a PLF. Although there might be promising quantitative approaches for quantifying the non-linear correlations, they have been considered out of scope for this work. The correlation graphs presented throughout this section have been designed by Yasin Bayzidi for the scientific paper version of this study [52]. The x-axis quantifies the value of the studied factor, while the left y-axis quantifies the detection performance with respect to PDSM. For scene-based factors that are computed at the image sample level, the F1-Score is used to measure the detection performance. Object-based factors, on the other hand, are linked to individual objects (pedestrians) that appear within the image samples and therefore do not support the tracking of FP detections, which are required to compute the precision and F1-Score. Because of this, object-based factors are evaluated based on the PDSM recall. The right y-axis is used to quantify the density histogram of the PLF values within the training splits. Furthermore, the orange and blue bars represent the density histograms of PLF values within the respective KI-A and CityPersons train splits. Finally, the dashed line plots represent the detection performance for each of the studied models over the range of all PLF values. The red line plots present the detection performance on KI-A, while the blue line plots represent the detection performance on CityPersons.

6.2.1 Valid Performance Limiting Factors (PLFs)

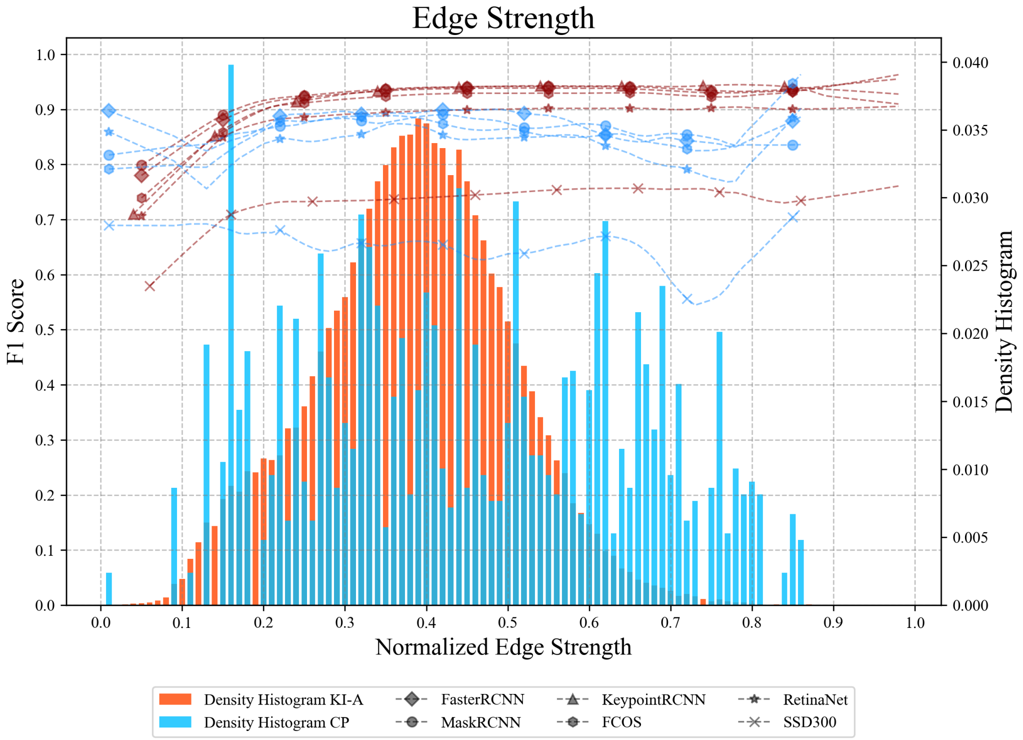


Figure 6.1: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **edge strength** factor.

The edge strength is a scene-based factor that quantifies the edge frequency within an image. A lower edge strength might suggest that the given image sample is blurry or appears to contain less information, while higher values indicate the opposite. There is a clear difference between the distribution of edge strength values within KI-A and CityPersons. The edge strength values within KI-A follow a normal distribution, whereas on CityPersons they follow a more or less uniform distribution between the values 0.1 and 0.8. Furthermore, as can be observed from the line plots, the overall edge strength of an image seems to have very little influence on the detection performance except for lower edge strength values ranging from 0 to 0.2. The studied models show a decrease in detection performance by more than 10% on the KI-A dataset within this area, and the detection performance on CityPersons also seems unstable for lower edge strength values. Since the edge strength can be easily computed for a given image, its value could be used to identify image samples on which a reduced detection performance is to be expected. Based on this correlation graph, the edge strength can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

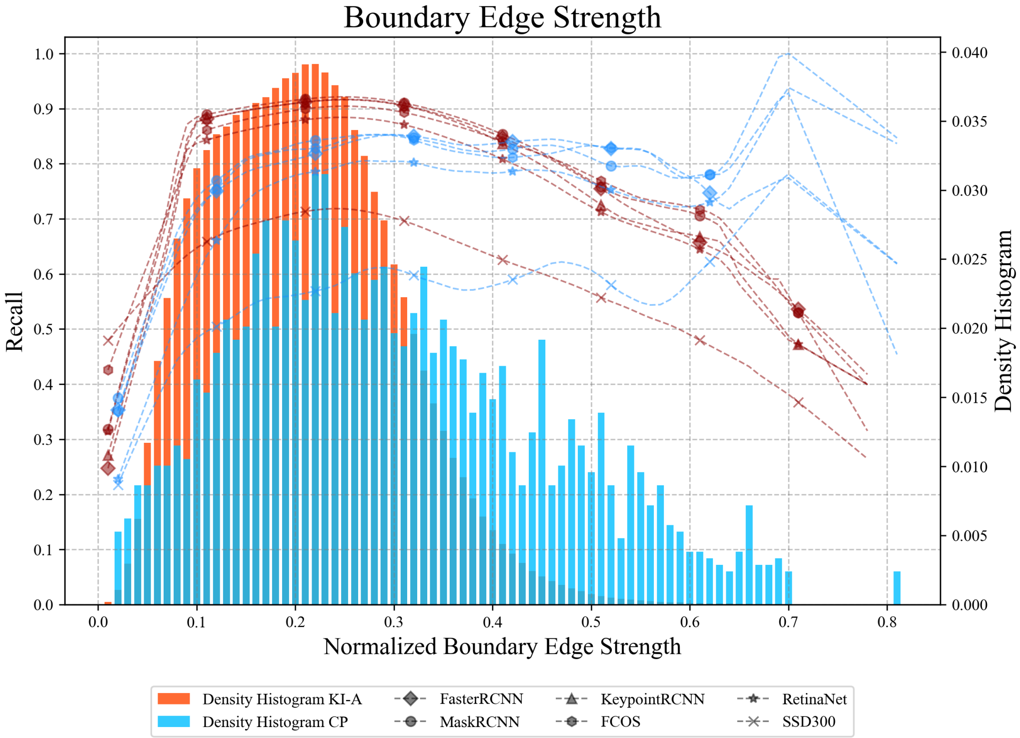


Figure 6.2: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the factor **boundary edge strength**.

The boundary edge strength is a novel object-based factor that was introduced as part of this work to quantify the edge frequency of the boundary separating the pedestrian from its background. Low values of this factor indicate a very smooth boundary between the pedestrian and its background, making it harder for the pedestrian detectors to identify each of the instances. Higher values, on the other hand, should indicate a very strong boundary and therefore increased visibility of the instances, making their detection easier. The distribution of factor values is similar within both datasets and appears skewed to the left. Furthermore, for a boundary edge strength spanning between 0 and 0.1, there is a 50% drop in detection performance on both datasets, while at the same time the distribution of factor values appears to be very high in this area. After the boundary edge strength value of 0.3, there is a clear negative performance trend on KI-A, whereas on CityPersons, a clear trend is difficult to identify due to a lack of data and the presence of outliers. This observation is contrary to the prior belief that higher boundary edge strength values would increase the detection performance; however, further experiments are required before conclusions on the effects of higher boundary edge strength values can be drawn. Based on this correlation graph, the boundary edge strength factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

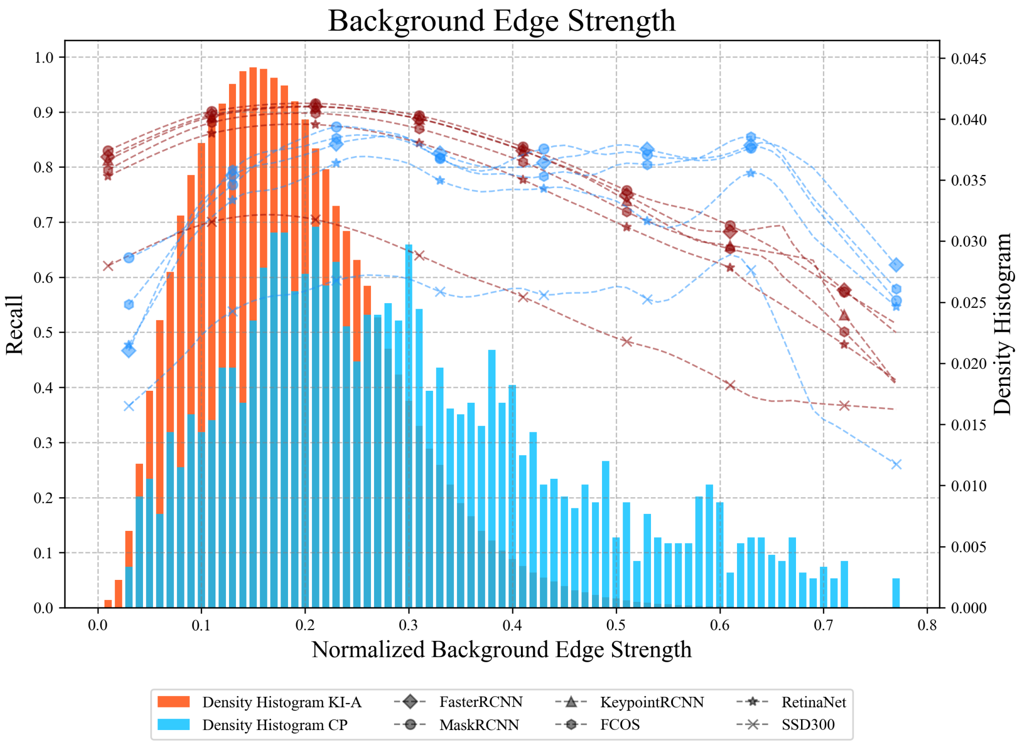


Figure 6.3: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **background edge strength** factor.

The background edge strength is yet another novel object-based factor that was introduced as part of this work to quantify the edge frequency of the bounding box background. A low background edge strength could manifest itself as a smooth background without any edges or a blurry part of the image. Higher values indicate the presence of textures and complex backgrounds that could potentially mislead the identification of pedestrian instances. The distribution of factor values within both datasets appears to be highly similar to the previous boundary edge strength factor. It can be observed that for very low background edge strength values ranging from 0 to 0.1, there is a significant performance decrease, even though the distribution of factor values is quite high in this area. This decrease in detection performance is most significant on CityPersons where the recall appears more than 30% below the baseline performance. Similarly to before, after the value of 0.3, the detection performance on KI-A drops sharply, whereas on CityPersons, the detection performance remains longer stable before also collapsing at the background edge strength of 0.63. Based on this correlation graph, the background edge strength factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

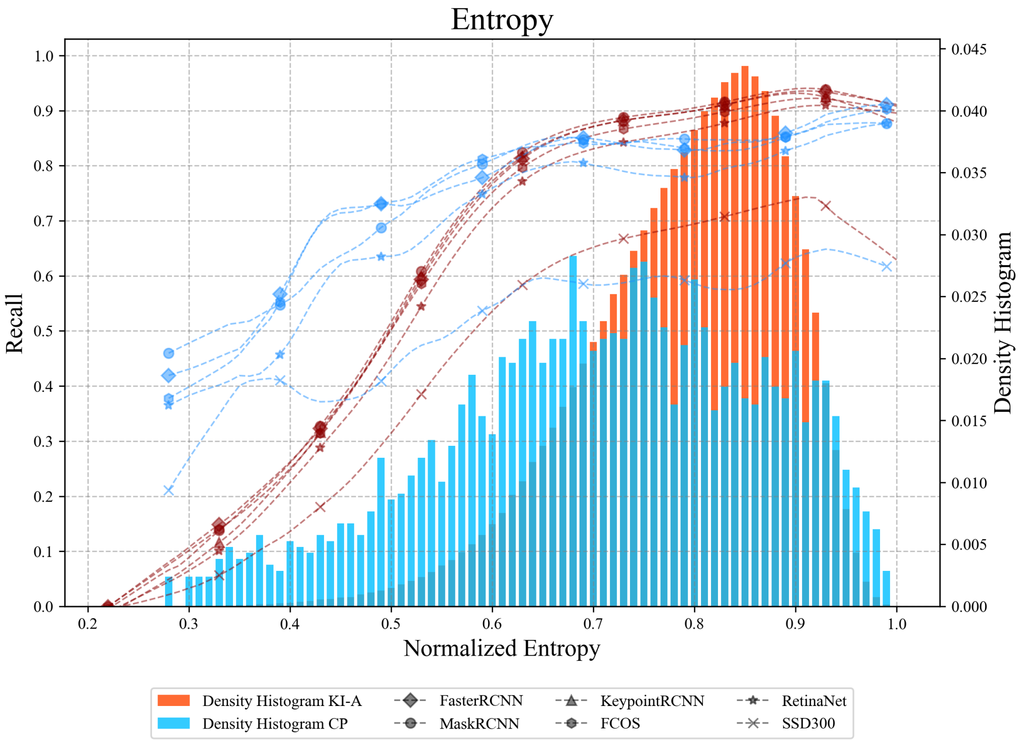


Figure 6.4: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **entropy** factor.

The entropy is an object-based factor responsible for tracking the Shannon entropy of a pedestrian bounding box image. Its ability to measure statistical randomness in the image data enables entropy to indicate the presence of textures and clear pedestrian boundaries. The factor values have a similar distribution within both datasets, which appears skewed to the right. As one would expect, the detection performance positively correlates with the value of the normalized entropy. This effect is more clearly identifiable on the KI-A dataset, where the effect of very low entropy values completely collapses the detection performance to 0%. Even though, it is clear that the number of training samples in this area is insufficient, nonetheless, low-entropy pedestrian instances should be investigated further to better understand their effects on detection performance. Furthermore, there are clear correlation trends on both datasets to be observed, which show that higher entropy values lead to higher detection rates, confirming the entropy factor as a valid Performance Limiting Factor (PLF) for DNNs.

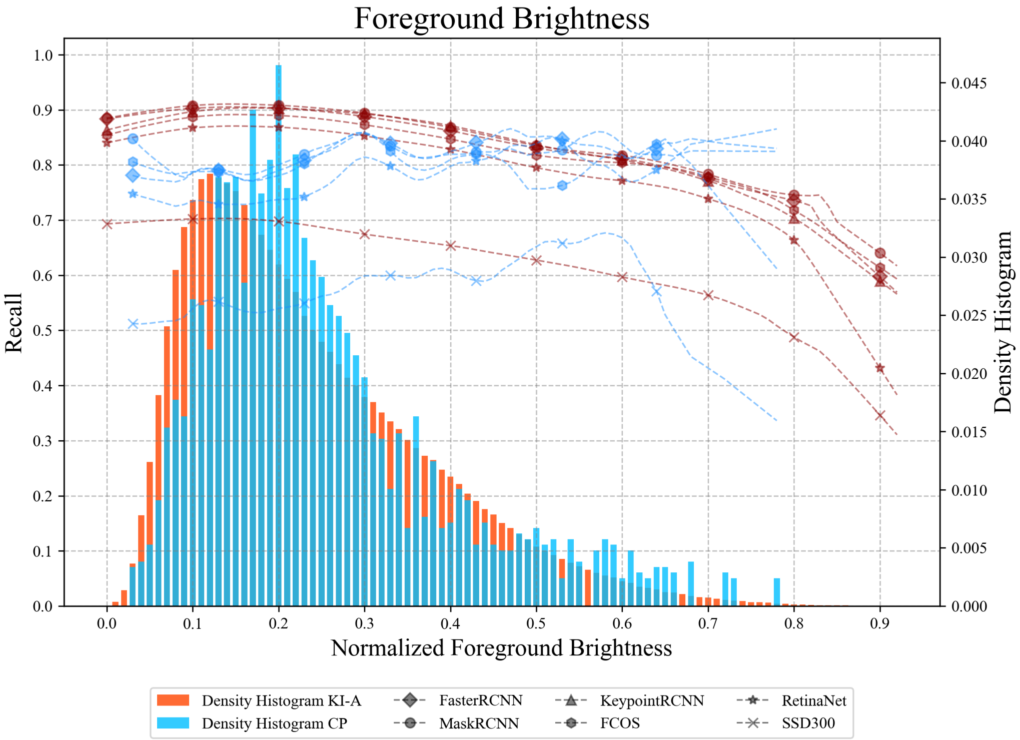


Figure 6.5: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **foreground brightness** factor.

The foreground brightness is an object-based factor used for tracking the overall pixel brightness of a pedestrian's bounding box foreground. Both datasets feature a very similar distribution of factor values, which appears skewed to the left. However, when it comes to the detection performance of the studied pedestrian detectors, there is an obvious divergence between the two datasets. It appears that higher foreground brightness values have a much higher impact within KI-A leading to a drop of over 30% in detection performance for the foreground brightness value of 0.9. On CityPersons, the same pedestrian detectors appear to be unaffected by the foreground brightness factor. It should be highlighted here that the CityPersons test split consists of only 500 image samples, making it statistically prone to noise and outliers, while the data stemming from KI-A is very accurately measured on more than 145,000 image samples from the test split. Therefore, based on the observed effects on KI-A the foreground brightness can still be considered a valid Performance Limiting Factor (PLF) for DNNs.

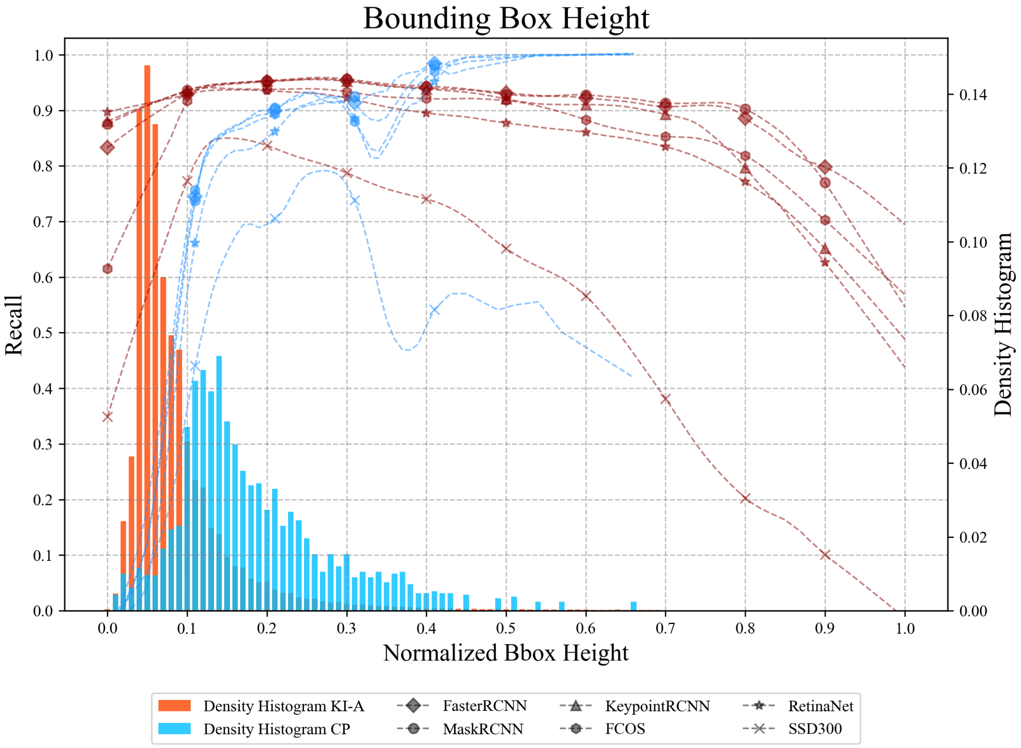


Figure 6.6: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **bounding box height** factor.

The bounding box height is an object-based factor that measures the pixel height for a given bounding box. The first notable observation is that the average bounding box within KI-A appears shorter compared to the average CityPersons bounding box. This is due to the CityPersons dataset having a higher image resolution, resulting in higher bounding box pixel heights. Besides this, the distribution of factor values appears very similar within both datasets, with very few instances having a very high bounding box height. Furthermore, it can be observed that the detection performance diminishes for very small factor values spanning between 0 and 0.1. As the value of the bounding box height grows, so does the detection performance. Unfortunately, due to a lack of training data, there are no verifiable results on the correlation between detection performance and higher values of this factor. Based on this correlation graph and the effects of lower values of this factor, the bounding box height factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

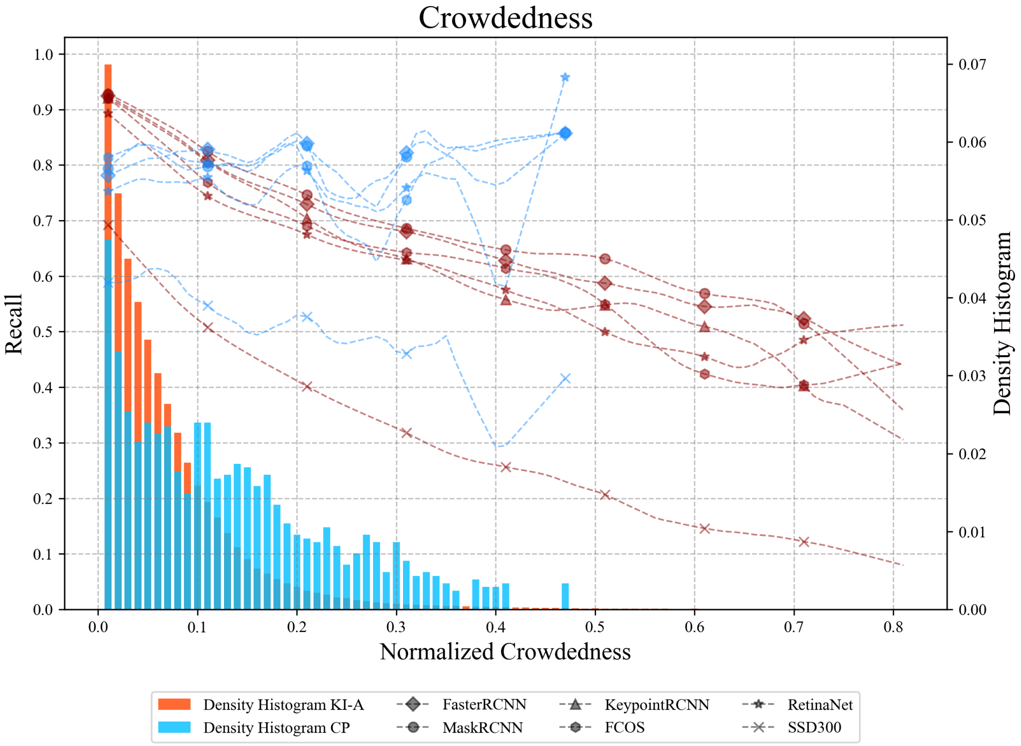


Figure 6.7: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **crowdedness** factor.

The crowdedness is a novel object-based factor that estimates the degree to which a pedestrian appears crowded within a given image. Its distribution of factor values is highly similar within both datasets, except for higher crowdedness values, which only appear in KI-A. This is not surprising given that KI-A contains image samples with densely packed pedestrian instances, resulting in higher individual crowdedness scores. The crowdedness value is shown to have a negative correlation with detection performance on KI-A, with a crowdedness value of 0.4 resulting in a more than 30% decrease in detection performance. The same effect cannot be clearly observed on CityPersons and requires further experiments to be fully verified. Based on the observed effects on KI-A the crowdedness can be considered a valid Performance Limiting Factor (PLF) for DNNs.

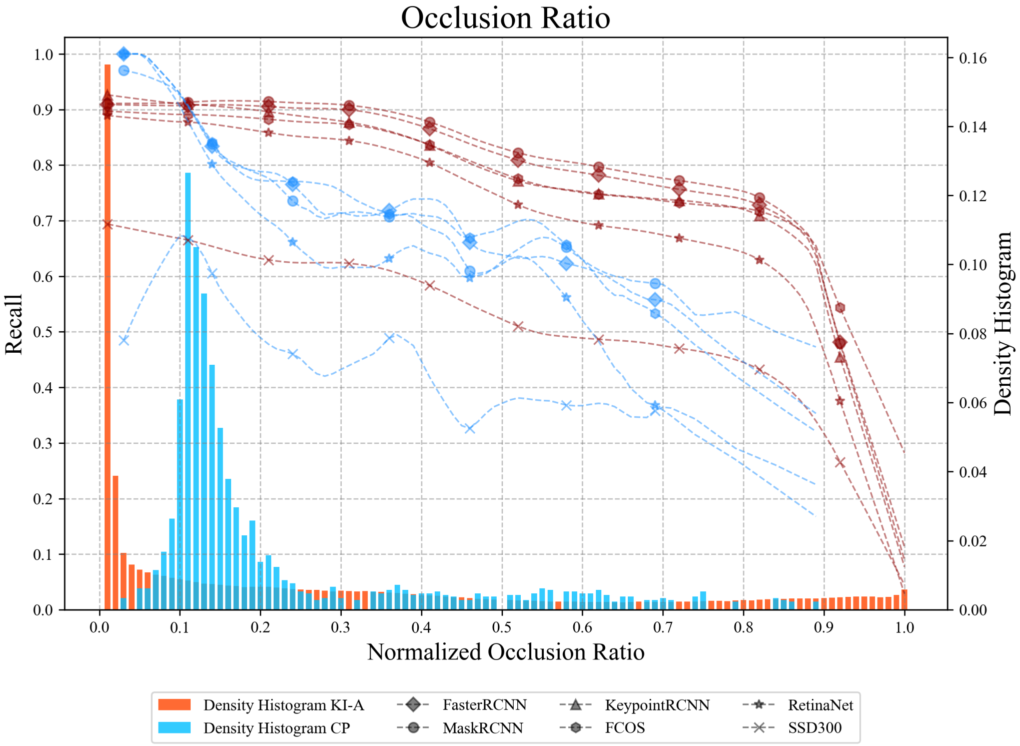


Figure 6.8: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **occlusion ratio** factor.

The occlusion ratio is an object-based factor that measures the ratio between the amount of visible instance pixels and the total amount of instance pixels without any occlusions. As a reminder, the occlusion ratio has been approximated for the CityPersons dataset by the occlusion estimation regression model, which explains why the distribution of factor values has its peak at around 0.11 while most of the pedestrian samples on KI-A appear unoccluded. Furthermore, it seems like most of the models are able to handle low occlusion levels on KI-A quite well. The first significant decrease in detection performance appears after the occlusion ratio of 0.4, and a total collapse in detection performance is observable after the occlusion ratio of 0.9. On CityPersons, the detection performance decreases more or less in a linear fashion with the increase in the occlusion ratio. Based on this correlation graph, the occlusion ratio factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

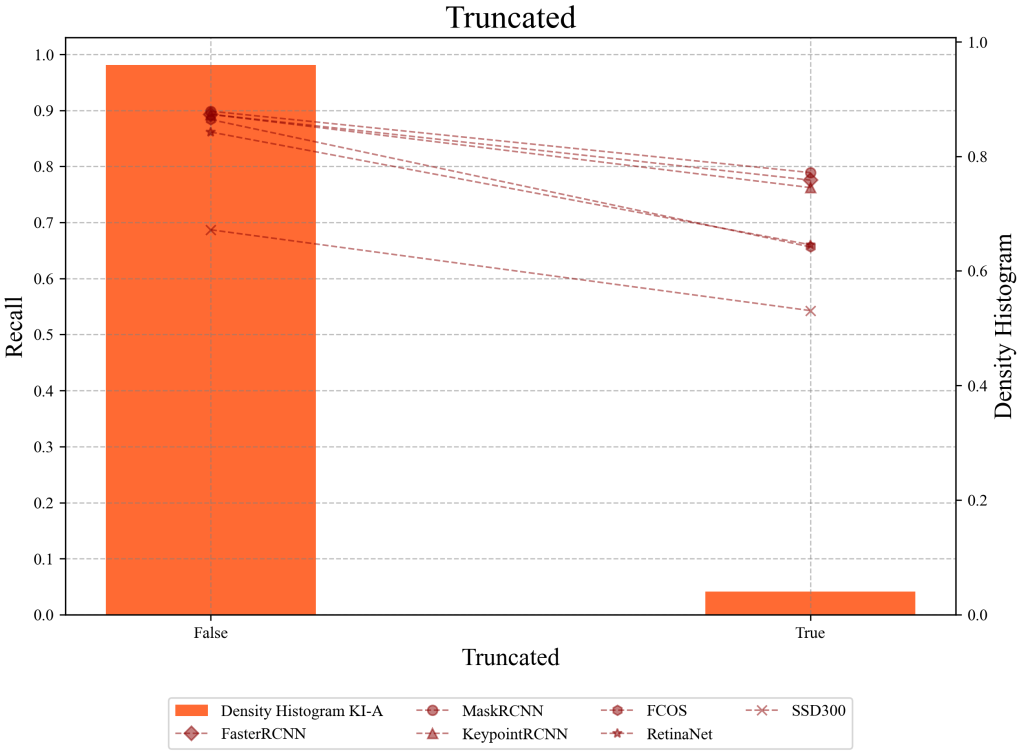


Figure 6.9: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **truncation** factor.

The truncation is a binary object-based factor whose purpose is to indicate whether the pedestrian's bounding box has been truncated as a consequence of being located outside the image's borders. This factor has only been tracked for the KI-A dataset and distinguishes between two possible values. The value of "False" indicates that a given bounding box is not truncated, while the value "True" marks a truncated pedestrian instance. Even though the amount of train samples that appear truncated is very low its effects should still be recognized since they act in a similar way like the occlusion factor. Furthermore, it can be clearly observed that the detection performance drops in the case that the pedestrian instance appears truncated. The interesting fact here is that the two-stage detectors Faster R-CNN, Mask R-CNN, and Keypoint R-CNN appear to be far more robust compared to the other models, since their detection performance drops by just 10% compared to the 20% drop in detection performance for the studied one-stage detectors. Based on this correlation graph and the background knowledge about the effects of this factor, the truncation factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

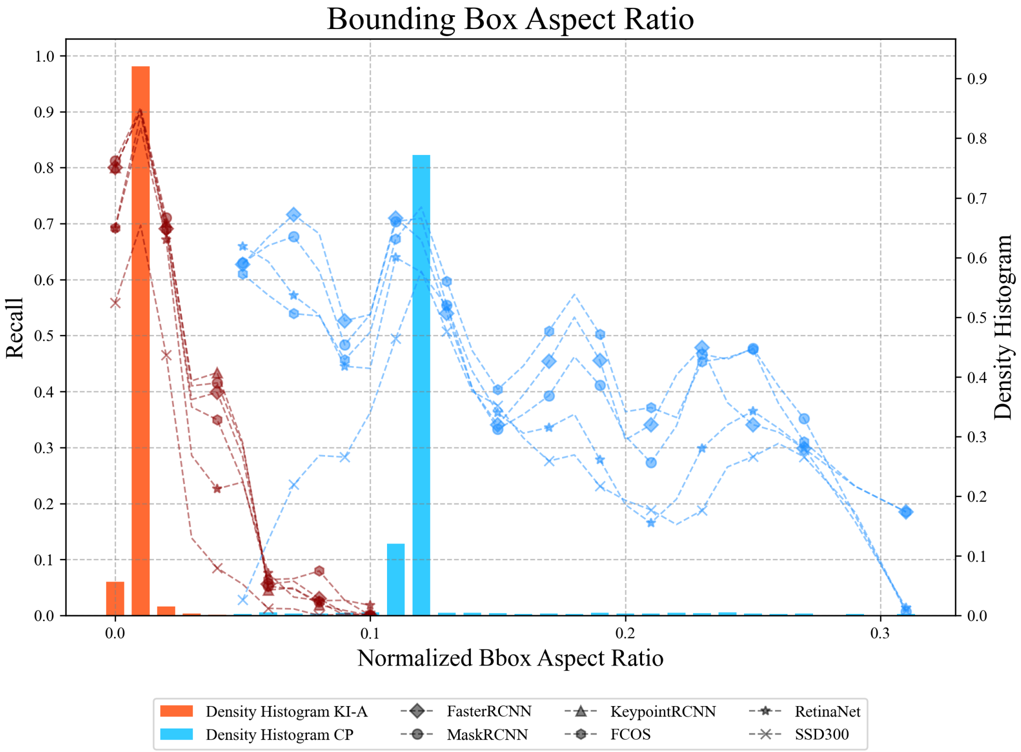


Figure 6.10: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **bounding box aspect ratio** factor.

The bounding box aspect ratio is an object-based factor that measures the ratio between the bounding box's pixel width and its pixel height. The distribution of factor values is heavily skewed towards a single value for both datasets. Consequently, this leads to a drop in detection performance for other bounding box aspect ratios, which can at least in part be attributed to the lack of training data with varying bounding box aspect ratios. At the same time, it is this lack of training data that makes this factor so sensitive for pedestrian detection. It can be assumed that a similar distribution of factor values can be observed on all pedestrian detection datasets that are used for optimizing pedestrian detectors, indicating that the main problem with varying bounding box aspect ratios is their underrepresentation within the training data. Future research efforts in the pedestrian detection domain should therefore investigate potential methods for improving the detection performance for varying bounding box aspect ratios by carefully curating the training datasets. Based on this correlation graph and the particularity of this factor, the bounding box aspect ratio factor can be considered a valid Performance Limiting Factor (PLF) for DNNs.

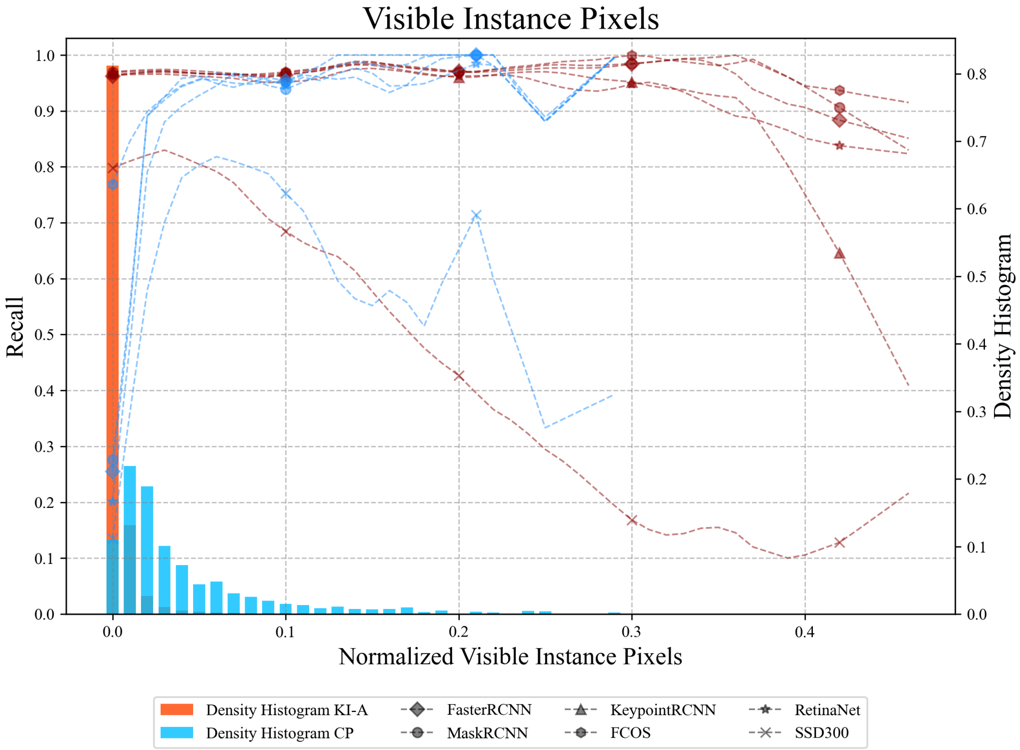


Figure 6.11: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **visible instance pixels** factor.

The visible instance pixels factor is an object-based factor that tracks the number of visible instance pixels for a given pedestrian instance. For both datasets, the distribution of factor values appears to be heavily skewed to the left, with very few training samples having higher amounts of visible instance pixels. With increasing values of this factor on KI-A, there is a very marginal performance increase to be observed, whereas pedestrian detectors on CityPersons appear to be more affected by lower values of this factor. There is also an interesting divergence between the individual models to be seen, with FCOS being far more robust to low amounts of visible instance pixels than all of the other CityPersons detectors studied in this work. Unfortunately, the data coverage for higher values of this factor is insufficient to draw any further conclusions about its effects on the detection performance. This, however, is a very crucial aspect since higher values of this factor represent pedestrian samples that are standing right in front of the camera within a very short range and cover larger portions of the input images. These properties should defintetely be adressed by future works to better understand their effects on the general detection performance. Based on this correlation graph, the visible instance pixels factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

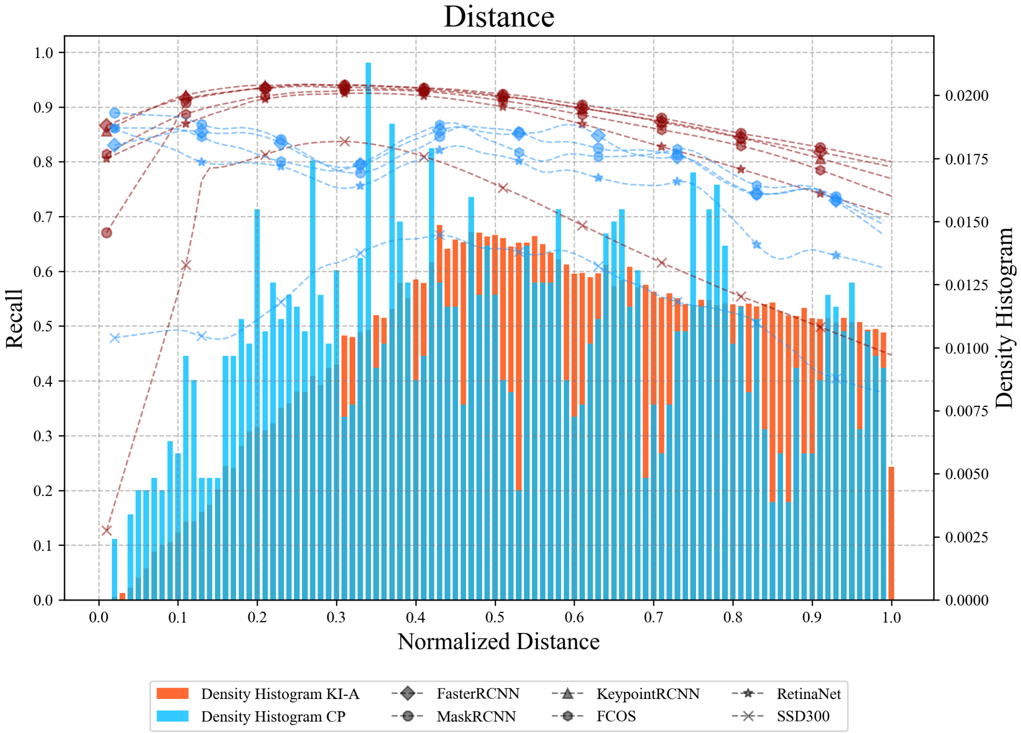


Figure 6.12: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **distance** factor.

The distance is an object-based factor that quantifies the range between a pedestrian instance and the camera. Its value has a significant impact on the visual appearance of a pedestrian instance within an image because it directly influences other object-based factors such as bounding box height and the amount of visible instance pixels. Both of the aforementioned factors are negatively correlated with the distance factor, meaning that with increasing distance, the bounding box height and the amount of visible instance pixels is decreasing. Since PDSM defines all pedestrians beyond a 50-meter distance threshold as non-safety-relevant, this graph shows the correlation between the detection performance and the distance factor spanning from 0 to 50 meters (normalized in the range from 0 to 1). The distribution of factor values appears almost uniform within both datasets. By inspecting the lower distance values, an interesting divergence in detection performance can be observed between the two datasets. While detectors on KI-A appear to have lower detection performance on pedestrians that are standing in close proximity to the camera, the detectors on CityPersons appear to perform best on this type of pedestrians. As the distance values increase, the detection performance decreases on both datasets, clearly indicating a decrease in the detection capabilities of the studied systems. Based on this correlation graph, the distance factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

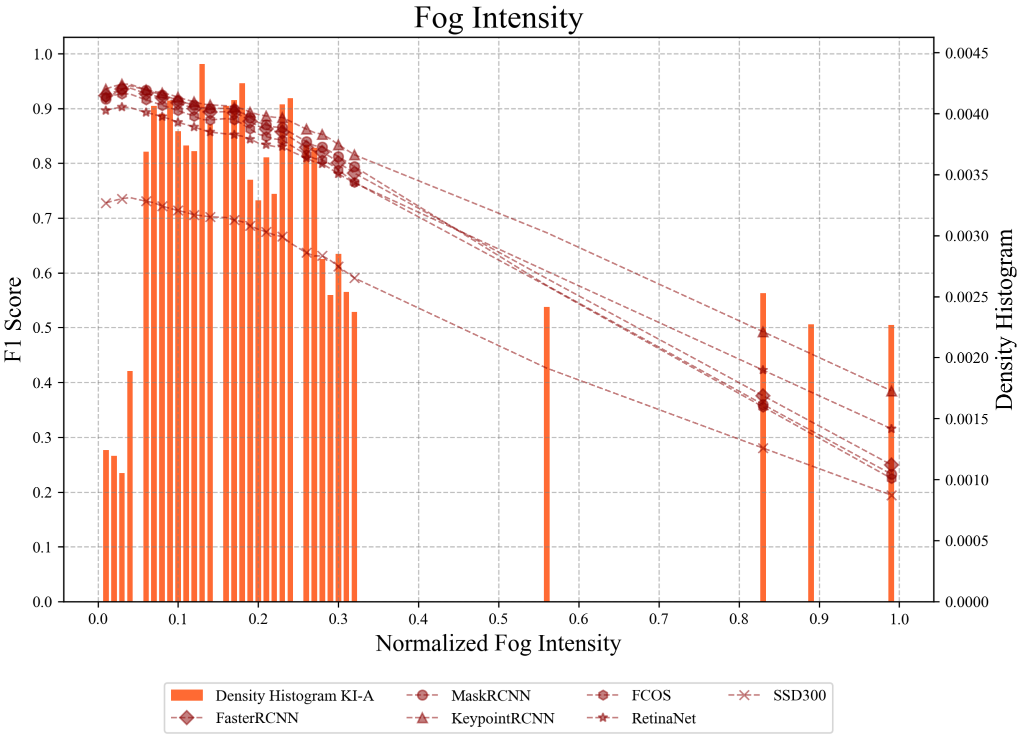


Figure 6.13: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **fog intensity** factor.

The fog intensity is a scene-based factor that quantifies the amount of fog that is added within the KI-A simulation. Its distribution of factor values is mostly skewed to the left, with several higher values appearing frequently. As one would expect, the detection performance of all studied pedestrian detectors decreases linearly with increasing fog intensity. This effect is so severe that higher fog intensity values reduce the detection capabilities by over 50%. Based on this correlation graph, the fog intensity factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

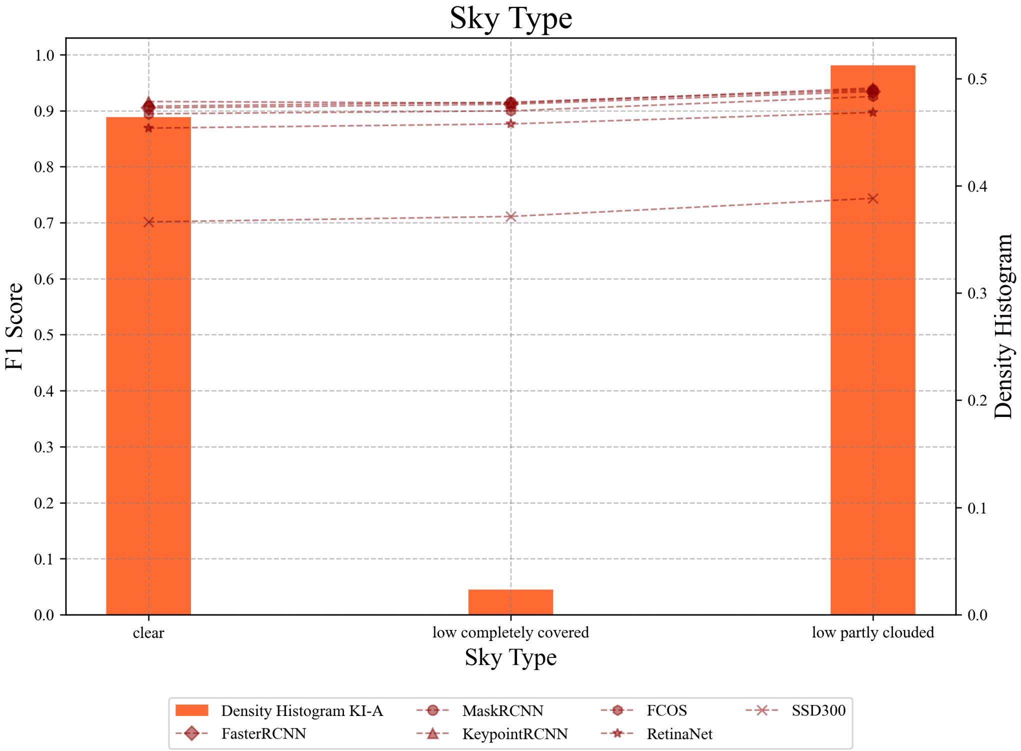


Figure 6.14: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **sky type** factor.

The sky type is a scene-based factor that describes the appearance of the sky in the form of three possible values: "clear", "low completely covered", and "low partly clouded". Its value has been tracked by the KI-A simulation and mostly remained "clear" or "low partly clouded" with fever training samples being marked as "low completely covered". Since the studied pedestrian detectors were optimized to detect pedestrian instances that appear on the ground, it was expected that the appereance of the sky would have a rather minor role on the general detection performance. However, as can be observed from the correlation graphs, it appears that the detection performance is lowest on image samples that contain a clear sky. Image samples with the sky type set to "low completely covered" show a minor increase in detection performance, even though they were much less frequent in the training data. Furthermore, image samples with a sky type of "low partly clouded" show a 5% increase in detection performance. These results indicate that there might be other factors that correlate with the sky type and therefore influence the detection performance. Based on this correlation graph, the sky type factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

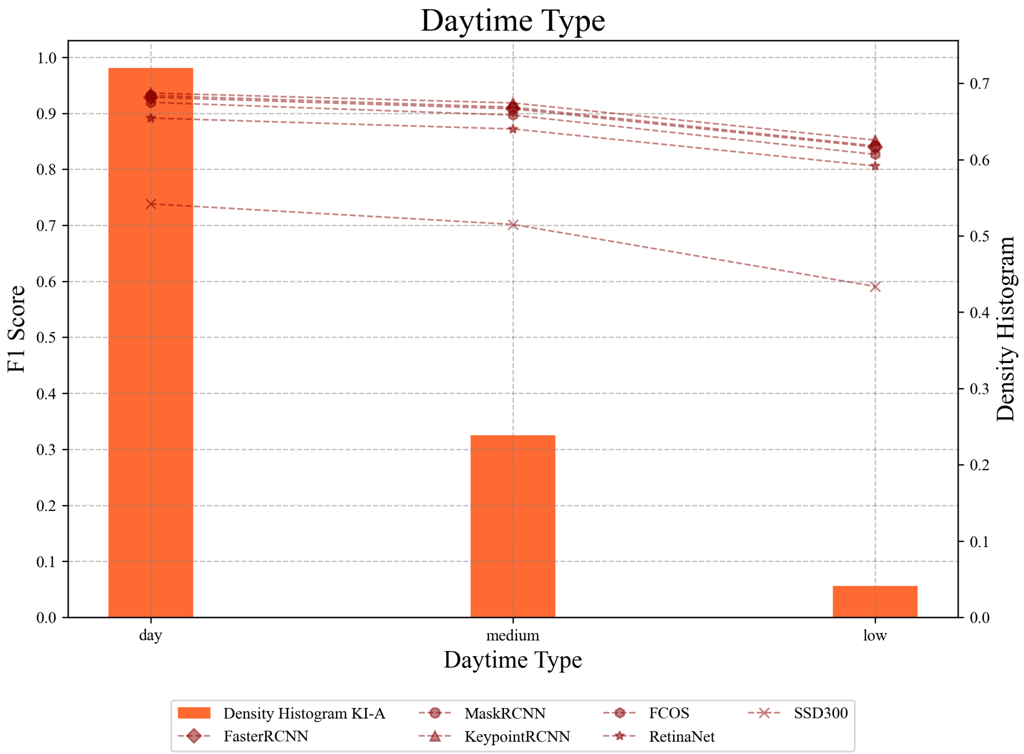


Figure 6.15: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **daytime type** factor.

The daytime type is yet another categorical scene-based factor that represents the daytime of a given image sample within the KI-A simulation in the form of three possible values, including "day", "medium", and "low". As can be clearly observed from the correlation graphs, the highest detection performance is achieved on image samples that are taken at daytime, represented by the category "day". Image samples that were taken at "medium" daytime, representing the beginning of sunset or the end of sunrise, show a minor decrease in detection performance. The last category, "low" daytime, represents image samples that were taken at the end of sunset or the beginning of sunrise, adding a stronger orange tone to the image samples, which also appear darker due to this effect. For these types of image samples, there is a 10% decrease in detection performance to be observed. These findings, along with the results of the sky type factor, demonstrate that future pedestrian detectors must also adapt to such environmental conditions that might influence the detection performance in unexpected ways. Based on this correlation graph, the daytime type factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

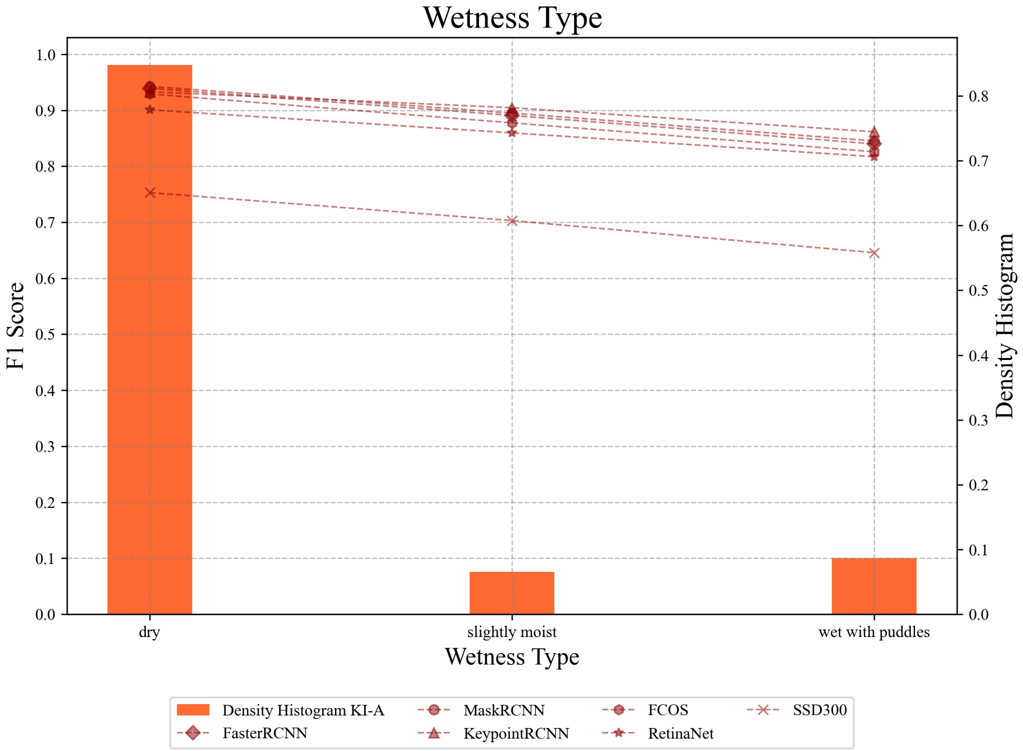


Figure 6.16: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **wetness type** factor.

The wetness type is a scene-based factor that describes the degree to which the ground appears wet in an image from KI-A. The three possible categories are: "dry", "slightly moist", and "wet with puddles". Even though most of the KI-A samples appear in dry conditions, over 10% of training samples contained slightly moist roads or roads that are wet with puddles. As can be observed from the graph above, it appears that wetness is a factor that negatively influences the detection performance, reducing the overall F1-Score by almost 10%. This observation is probably due to the visual effect of wetness causing reflections and light artifacts in the image, making the identification of pedestrian instances much harder than in dry conditions. Based on this correlation graph, the wetness type factor can be confirmed as a valid Performance Limiting Factor (PLF) for DNNs.

6.2.2 Invalid Performance Limiting Factors (PLFs)

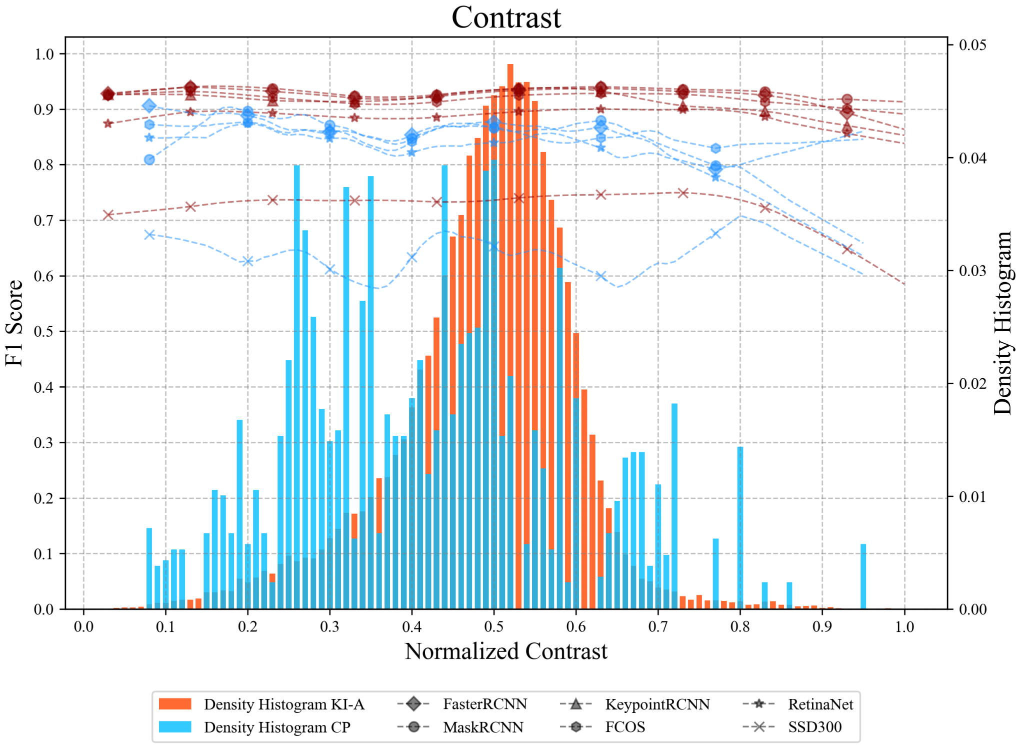


Figure 6.17: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **contrast** factor.

The contrast is a scene-based factor responsible for tracking the contrast of a given image. Its value serves as a good indication about the image quality, with lower values indicating that the distribution of pixel intensity values is skewed, resulting in images with reduced visibility to the human eye. High values represent a more even distribution of pixel intensity values within an image and, therefore, increased visibility. Similarly to the edge strength factor, the distribution of contrast values within KI-A appears to follow a normal distribution, while within CityPersons, the values are more evenly distributed over the whole range. Furthermore, there is no clear effect observable on the detection performance except for higher contrast values. However, since the amount of training data is very low in this area, its effect can be neglected. Based on this correlation graph, the contrast factor can be rejected as a PLF for DNNs.

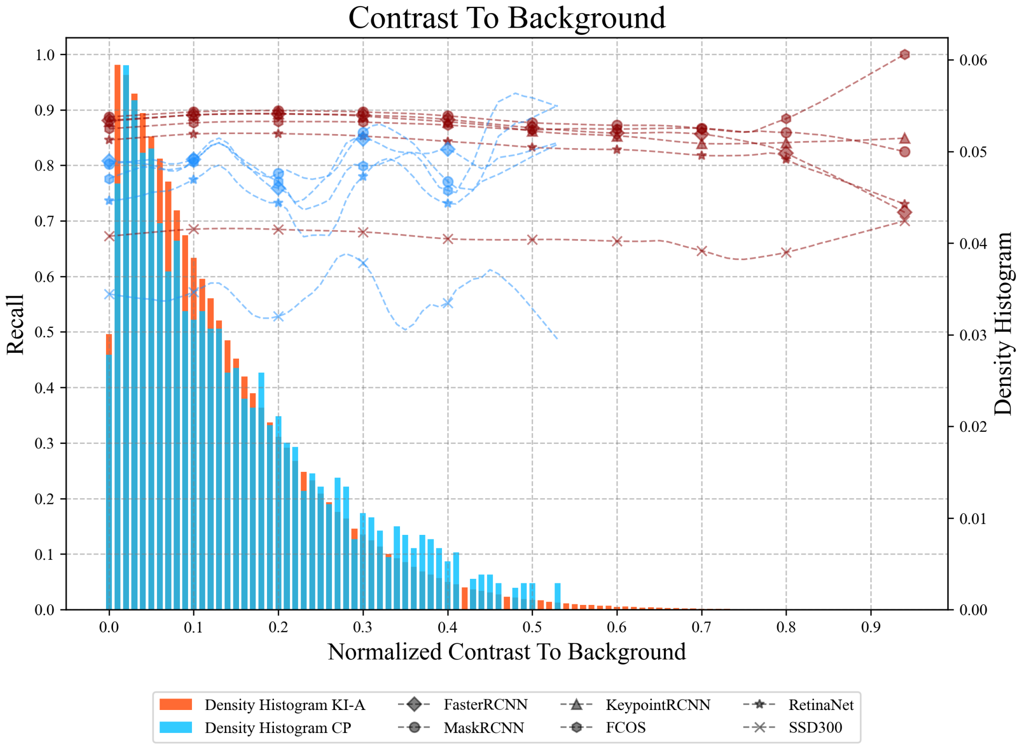


Figure 6.18: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **contrast to background** factor.

The contrast to background is a novel object-based factor that quantifies the difference in contrast between foreground and background of a pedestrian bounding box. Intuitively, a higher contrast to background should occur in cases where the pedestrian instance is clearly visible and separable from its background area, making it easier for the pedestrian detectors to detect it. As can be observed from the density histograms, the distribution of contrast to background values is highly skewed to the left on both datasets. This finding suggests that the contrast between the bounding box foreground and background is highly similar most of the time. Surprisingly, there seems to be no effects on the detection performance with respect to the contrast to background factor. Moreover, as the value of this factor grows so does the amount of training samples decrease making any further observable effects invalid. Based on this correlation graph, the contrast to background factor can be rejected as a PLF for DNNs.

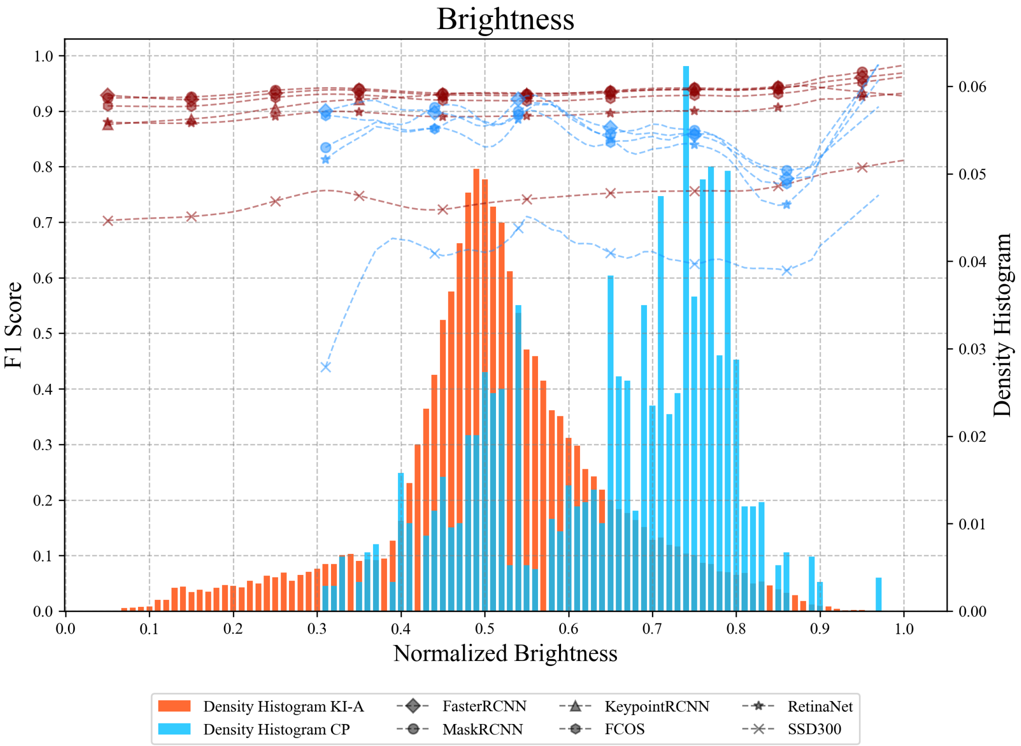


Figure 6.19: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **brightness** factor.

The brightness is a scene-based factor responsible for tracking the overall pixel brightness of a given image. Similarly to the edge strength and contrast factors, the distribution of brightness values within KI-A appears to follow a normal distribution, while within CityPersons, the values are more skewed to the right. Judging by the line plots, there is a minor increase in detection performance observable with increasing brightness values; however, the distribution of training data makes this claim invalid, and further experiments would be required to verify whether there is a correlation with the detection performance. Therefore, based on this correlation graph, the brightness factor can be rejected as a PLF for DNNs.

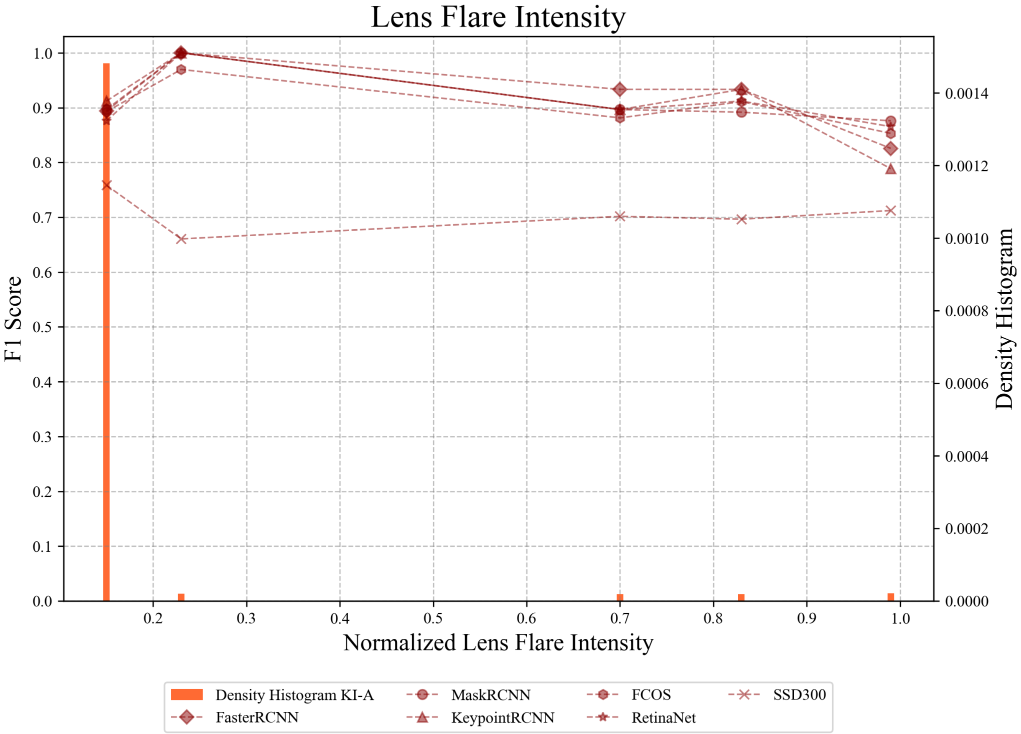


Figure 6.20: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **lens flare intensity** factor.

The lens flare intensity is a scene-based factor that quantifies the intensity of lens flare artefacts that are added to the image samples by the KI-A simulation. These artifacts could pose a risk to the reliability of modern pedestrian detectors, leading to a drop in detection performance. The correlation graph clearly shows that the data coverage for image samples with lens flare artifacts is insufficient to extract meaningful information about their influence on the detection performance. Nonetheless, image samples with lens flare artifacts show a decrease in detection performance in three of the four factor values that are observable from the graph. This could be due to the lens flare artefact overlapping pedestrian instances leading to FNs, or it could be due to the lens flare artefacts causing potential missdetection, resulting in FPs, which reduces overall precision and F1-Score. In order to fully grasp and understand the effects of lens flare artifacts, further experiments are required. Until then, the lens flare intensity factor will be rejected as a PLF for DNNs.

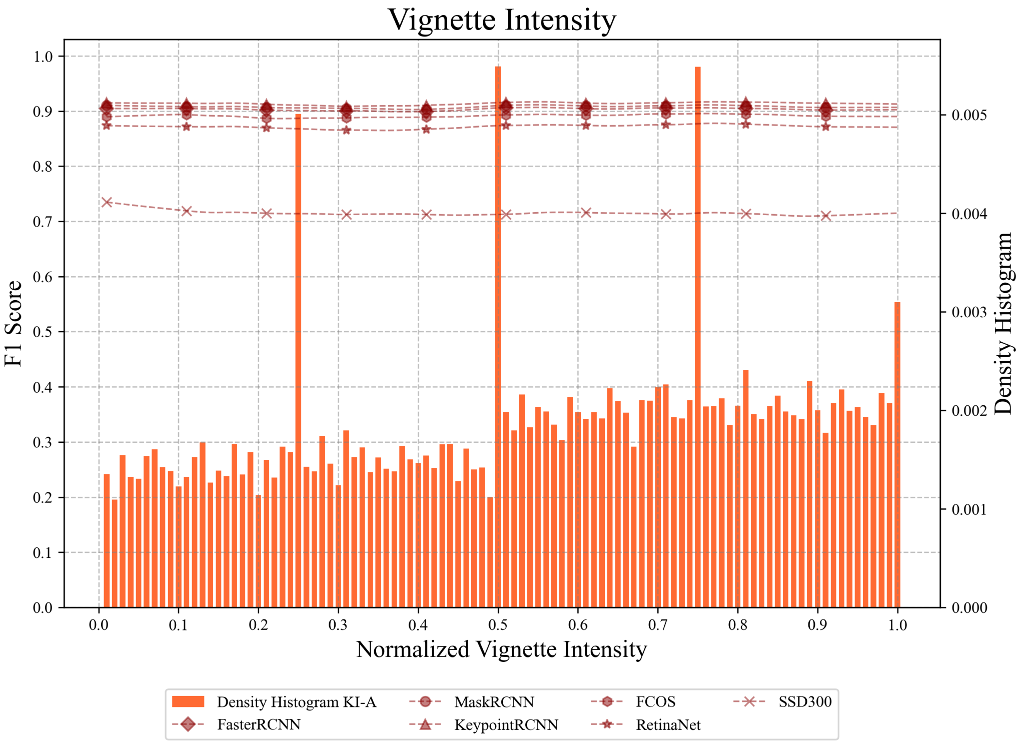


Figure 6.21: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **vignette intensity** factor.

The vignette intensity is yet another scene-based factor that quantifies the intensity of the vignetting effect, leading to lower pixel intensity values around the image's borders by making them appear darker. This factor is only tracked for the KI-A dataset since it belongs to the group of meta annotations. The distribution of factor values appears more or less uniform, as does the detection performance for all of the studied pedestrian detectors. It would be interesting to investigate whether an unbalanced distribution of factor values within the training data could lead to a drop in detection performance with respect to this factor. Based on this correlation graph, the vignette intensity factor can be rejected as a PLF for DNNs.

Chapter 7

Conclusion

7.1 Summary

This section summarizes the main contributions of this work and the key findings obtained from the results of this research. The main research objective of this work was to conduct a systematic safety analysis in order to identify the various factors that lead to DNN failure within the use-case of pedestrian detection for autonomous driving in urban traffic.

The main contributions of this work can be summarized as follows:

- This work introduced a novel and task-oriented Pedestrian Detection Safety Metric (PDSM) for assessing the detection performance of pedestrian detectors in the context of autonomous driving within urban traffic.
- The concept of Performance Limiting Factors (PLFs) was introduced, and a first definition has been formulated to serve as argumentative guidance throughout this work.
- A total of 21 potential PLFs have been introduced. Out of these 21 factors, 17 were taken from the literature, while the other four factors have been introduced as novel factors.
- The newly introduced factors include methods for quantifying the boundary edge strength, the background edge strength, the crowdedness, and the contrast to background for a given pedestrian instance.
- To replace the missing occlusion annotations within CityPersons, this work introduced a novel occlusion estimation regression model that utilizes the full body bounding box and the binary instance segmentation mask to estimate the occlusion ratio for a given pedestrian instance.
- By utilizing the developed occlusion estimation regression model, this study further enriched the CityPersons dataset by extending each pedestrian ground truth annotation with its respective occlusion ratio information.

- In order to assess the pedestrian detectors at their highest level of performance, this study introduced a novel training approach for F1-Score maximization.

- Six state-of-the-art pedestrian detectors covering three detection tasks, including 2D-OD, 2D-IS, and 2D-KD have been trained on the synthetic KI-A and real-world CityPersons datasets, and their final detection performance has been reported with respect to the newly introduced PDSM and other standard 2D-OD metrics from the literature.

- Each of the 21 factors has been assessed based on correlation graphs that visualize the correlation between the detection performance and various values of the studied factors. Furthermore, the distribution of factor values within the train splits has been visualized to ease the qualitative analysis of each individual factor and help determine whether it fulfills the requirements for a PLF. Based on this qualitative analysis, 16 of the initial 21 factors have been confirmed as PLFs for DNNs.

A final summary of the PLF analysis is given in Table 7.1. The aforementioned table contains information on whether the PLF requirements have been fulfilled and qualitatively observed within the KI-A and CityPersons datasets for each of the 21 studied factors. This study identified four distinct PLFs that highly correlate with the detection performance within both datasets. These four factors include **entropy**, **occlusion ratio**, **boundary edge strength**, and **bounding box aspect ratio**. It should be highlighted here that the boundary edge strength is a novel factor introduced by this work and therefore represents a valuable contribution alongside the **background edge strength** and **crowdedness** factors, which are also novel factors that have been confirmed as PLFs. Furthermore, the factors **lens flare intensity**, **vignette intensity**, **contrast**, **contrast to background**, and **brightness** have been rejected as PLFs for DNNs, based on the qualitative assessments of the respective correlation graphs. However, this should not be regarded as proof that these factors do not impact the general 2D-OD performance. Moreover, there were no noteworthy effects observable with respect to the hybrid models Mask R-CNN and Keypoint R-CNN, which support 2D-IS and 2D-KD in addition to the general 2D-OD task. These models achieved similar detection performance to the baseline Faster R-CNN detector in most of the correlation graphs, which is contrary to the prior belief that such hybrid models would achieve higher robustness towards PLFs. Nonetheless, the studied two-stage detectors demonstrated once more their superior detection performance over the studied one-stage detectors by being more robust towards multiple PLFs. Appendix B concludes the results of this study by presenting the correlation graphs of another two factors that were not considered in the main study but still offer valuable insights into the DNN's behavior.

Factors	PLF_{KI-A}	$PLF_{CityPersons}$	Sev_{KI-A}	$Sev_{CityPersons}$
Entropy	Yes	Yes	High	High
Occlusion Ratio	Yes	Yes	High	High
Boundary Edge Strength*	Yes	Yes	High	High
Bounding Box Aspect Ratio	Yes	Yes	High	High
Bounding Box Height	Yes	Yes	Medium	High
Fog Intensity	Yes	-	High	-
Visible Instance Pixels	No	Yes	None	High
Background Edge Strength*	Yes	Yes	Medium	Medium
Distance	Yes	Yes	Medium	Medium
Truncation	Yes	-	Medium	-
Crowdedness*	Yes	No	Medium	None
Edge Strength	Yes	Yes	Low	Low
Foreground Brightness	Yes	No	Low	None
Sky Type	Yes	-	Low	-
Daytime Type	Yes	-	Low	-
Wetness Type	Yes	-	Low	-
Lens Flare Intensity	No	-	None	-
Vignette Intensity	No	-	None	-
Contrast	No	No	None	None
Contrast to Background*	No	No	None	None
Brightness	No	No	None	None

Table 7.1: Summary of the final Performance Limiting Factor (PLF) analysis. The first column lists all 21 investigated factors, sorted by their impact on detection performance. The second and third columns indicate whether the PLF requirements have been fulfilled by the respective factors on the KI-A and CityPersons datasets. The possible values are "Yes", "No" or "-", which is used in the case that the factor has only been tracked by one of the two datasets. The final fourth and fifth columns indicate the severity of the respective factors on the detection performance within KI-A and CityPersons. The possible values are categorized as "None", "Low", "Medium", "High", or "-", which is again used in the case that the factor has only been tracked by one of the two datasets.

*Novel Factor.

7.2 Future Work

This work presented a systematic safety analysis of 2D-OD pedestrian detectors by investigating the properties of PLFs and their effects leading to DNN failure. There are several interesting research directions that could yield promising scientific contributions based on further extensions of this work. One such extension of this study could define a metric for measuring the non-linear correlation between the PLFs and the detection performance. A promising approach for defining such a metric would include using the individual data points from the correlation graphs to fit a polynomial term that approximates the data (also known as "curve fitting"). To quantify the non-linear correlation, one would simply compute the Root Mean Squared Error (RMSE) between each data point from the correlation graph and the fitted polynomial curve. For PLFs that highly correlate with the detection performance, there would be a lower RMSE measured since the polynomial curve would better approximate the distribution of performance values. For factors that have a lower correlation with the detection performance, it is expected that the polynomial curve fitting will be much less accurate, resulting in a much higher RMSE. Furthermore, by defining some threshold values for the measured RMSE, one could more precisely categorize the severity of each PLF on the detection performance, thereby extending this study from a qualitative to a quantitative PLF analysis.

Another intriguing extension of this work could look into ways to utilize the knowledge of the DNN's limitations to further robustify them. By measuring the detection performance over several PLF values, one could use this data to identify the pedestrian instances that are more challenging for the detectors. Based on this concept, a new metric can be defined that combines all PLF effects into a single "difficulty" metric. This difficulty metric could then be incorporated into the training loss functions of the pedestrian detectors to assign a higher weight to misdetections of pedestrian instances with higher difficulty. Finally, a new training curriculum for DNNs could be introduced that focuses on strengthening the main weak spots of 2D-OD detectors.

This study encourages future works to continue the investigation of PLFs for DNNs by introducing further potential factors and by extending this study towards other use-cases.

Bibliography

[1] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles j3016_202104," https://www.sae.org/standards/content/j3016_202104/, accessed: 2022-12-04.

[2] D. Shin, K.-m. Park, and M. Park, "High definition map-based localization using adas environment sensors for application to automated driving vehicles," *Applied Sciences*, vol. 10, no. 14, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/14/4924

[3] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3213–3221.

[4] "Mean average precision (map) explained: Everything you need to know," https://www.v7labs.com/blog/mean-average-precision, accessed: 2022-12-06.

[5] "Synthetic data generation based on a modern game engine," https://www.ki-absicherung-projekt.de/fileadmin/KI_Absicherung/Final_Event/KIA_rollup_28.pdf, accessed: 2022-11-07.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

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

[9] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[10] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[11] T. Fingscheidt, H. Gottschalk, and S. Houben, "Deep neural networks and data for automated driving: Robustness, uncertainty quantification, and insights towards safety," 2022.

[12] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *European conference on computer vision*. Springer, 2012, pp. 340–353.

[13] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 2018.

[14] A. Von Bernuth, G. Volk, and O. Bringmann, "Simulating photo-realistic snow and fog on existing images for enhanced cnn training and evaluation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 41–46.

[15] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the safety of machine learning for pedestrian detection at crossings," in *Computer Safety, Reliability, and Security: 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, September 16–18, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 197–212. [Online]. Available: https://doi.org/10.1007/978-3-030-54549-9_13

[16] M. Lyssenko, C. Gladisch, C. Heinzemann, M. Woehrle, and R. Triebel, "From evaluation to verification: Towards task-oriented relevance metrics for pedestrian detection in safety-critical domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 38–45.

[17] Y. Bayzidi, A. Smajic, F. Hüger, R. Moritz, S. Varghese, P. Schlicht, and A. Knoll, "Traffic sign classifiers under physical world realistic sticker occlusions: A cross analysis study," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 644–650.

[18] "Self-driving car," https://en.wikipedia.org/wiki/Self-driving_car, accessed: 2022-12-04.

[19] "How self-driving cars work: Sensor systems," https://www.udacity.com/blog/2021/03/how-self-driving-cars-work-sensor-systems.html, accessed: 2022-12-04.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[21] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1904–1912.

[22] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1259–1267.

[23] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?" in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6034–6043.

[24] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," *BMVC Press*, 2009.

[25] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," 2017. [Online]. Available: https://arxiv.org/abs/1711.07752

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[28] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "Eurocity persons: A novel benchmark for person detection in traffic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1844–1861, 2019.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[30] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1605.06409

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[32] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[33] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *European conference on computer vision.* Springer, 2020, pp. 665–681.

[34] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16. [Online]. Available: https://proceedings.mlr.press/v78/dosovitskiy17a.html

[35] I. Hasan, S. Liao, J. Li, S. U. Akram, and L. Shao, "Pedestrian detection: Domain generalization, cnns, transformers and beyond," 2022. [Online]. Available: https://arxiv.org/abs/2201.03176

[36] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," 2017. [Online]. Available: https://arxiv.org/abs/1712.00726

[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[38] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 606–613.

[39] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, "Foveation-based mechanisms alleviate adversarial examples," *arXiv preprint arXiv:1511.06292*, 2015.

[40] A. Nussberger, H. Grabner, and L. Van Gool, "Robust aerial object tracking in images with lens flare," in *2015 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2015, pp. 6380–6387.

[41] "Lens flare," https://en.wikipedia.org/wiki/Lens_flare, accessed: 2022-11-07.

[42] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[43] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018.

[44] K. Saad and S.-A. Schneider, "Camera vignetting model and its effects on deep neural networks for object detection," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–5.

[45] C. Berghoff, P. Bielik, M. Neu, P. Tsankov, and A. von Twickel, "Robustness testing of AI systems: A case study for traffic sign recognition," in *IFIP Advances in Information and Communication Technology.* Springer International Publishing, 2021, pp. 256–267. [Online]. Available: https://doi.org/10.1007%2F978-3-030-79150-6_21

[46] T. Hess, M. Mundt, I. Pliushch, and V. Ramesh, "A procedural world generation framework for systematic evaluation of continual learning," 2021. [Online]. Available: https://arxiv.org/abs/2106.02585

[47] P. Fischer and A. Smajic, "Towards explainable ai systems for traffic sign recognition and deployment in a simulated environment," *Academia.edu*, 2021.

[48] A. Smajic, "Entwicklung und erprobung eines interaktiven 3d-stadtmodells am beispiel des personennahverkehrsnetzwerks der stadt frankfurt," *Universitätsbibliothek Johann Christian Senckenberg*, 2020.

[49] I. Pliushch, M. Mundt, N. Lupp, and V. Ramesh, "When deep classifiers agree: Analyzing correlations between learning order and image statistics," in *European Conference on Computer Vision.* Springer, 2022, pp. 397–413.

[50] "Transport road safety speed limits," https://ec.europa.eu/transport/road_safety/going_abroad/germany/speed_limits_en.htm, accessed: 2022-11-07.

[51] "Calculating stopping distance: Braking is not a matter of luck," https://mobilityblog.tuv.com/en/calculating-stopping-distance-braking-is-not-a-matter-of-luck/, accessed: 2022-11-07.

[52] Y. Bayzidi, A. Smajic, J. D. Schneider, F. Hüger, and A. Knoll, "Performance limiting factors of deep neural networks for pedestrian detection," *33rd British Machine Vision Conference (BMVC)*, 2022.

[53] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[54] "Sobel derivatives," https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html, accessed: 2022-11-16.

[55] "Mathematical morphology," https://en.wikipedia.org/wiki/Mathematical_morphology, accessed: 2022-11-16.

[56] "Histograms - 2: Histogram equalization," https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html, accessed: 2022-11-16.

[57] "Entropy," https://www.mathworks.com/help/images/ref/entropy.html, accessed: 2022-11-16.

[58] "Shannon entropy," https://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.shannon_entropy, accessed: 2022-11-16.

[59] "Ki absicherung: Safe ai for automated driving," https://www.ki-absicherung-projekt.de/en/project, accessed: 2022-11-21.

[60] "Technology: Synthetic data for training and validating the ai function," https://www.ki-absicherung-projekt.de/en/technology, accessed: 2022-11-21.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[63] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[64] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[67] "Torchvision," https://pytorch.org/vision/stable/index.html, accessed: 2022-12-09.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner,

L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019. [Online]. Available: https://arxiv.org/abs/1912.01703

[69] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[70] “Fiftyone,” https://voxel51.com/fiftyone/, accessed: 2022-12-13.

[71] “Mongodb,” https://www.mongodb.com/, accessed: 2022-12-13.

Appendix A

Supplementary Training Results Figures

This supplementary chapter contains a collection of plots that visualize the training convergence, the detection performance as measured by PDSM over varying confidence thresholds, and the Recall-IoU curve for each of the studied detectors and datasets.

The first type of plots are the training convergence plots. The x-axis represents the number of epochs, while the left y-axis is responsible for quantifying the detection performance with respect to PDSM, and the confidence threshold. The right y-axis is used to quantify the value of the prediction loss on the train split. The blue, orange, and green line plots represent the respective precision, recall, and F1-Score on the validation split at different epochs. The black dashed line represents the values of the model's confidence thresholds at which the detection performance on the validation split has been maximized. Moreover, the blue, yellow, and green star symbols are positioned on the x-axis with respect to the best performing epoch on the validation split and quantify on the left y-axis the final test split detection performance with respect to precision, recall, and F1-Score. The dashed red line represents the respective loss value during the model's training on the train split over all epochs.

The second type of plots visualizes the effects of varying confidence thresholds on the PDSM performance for the test split of a given dataset. The x-axis represents the confidence thresholds, ranging from 0 to 1. The y-axis quantifies the PDSM detection performance. The blue, yellow, and green line plots represent the respective precision, recall, and F1-Score over varying confidence thresholds, while the dashed black line represents the best-performing confidence threshold, obtained from the evaluation on the validation split and used for the final evaluation on the test split.

The third type of plots visualizes the Recall-IoU curve on the test split of a given dataset. The blue line plot illustrates the respective recall value (quantified by the y-axis) at varying IoU thresholds (quantified by the x-axis).

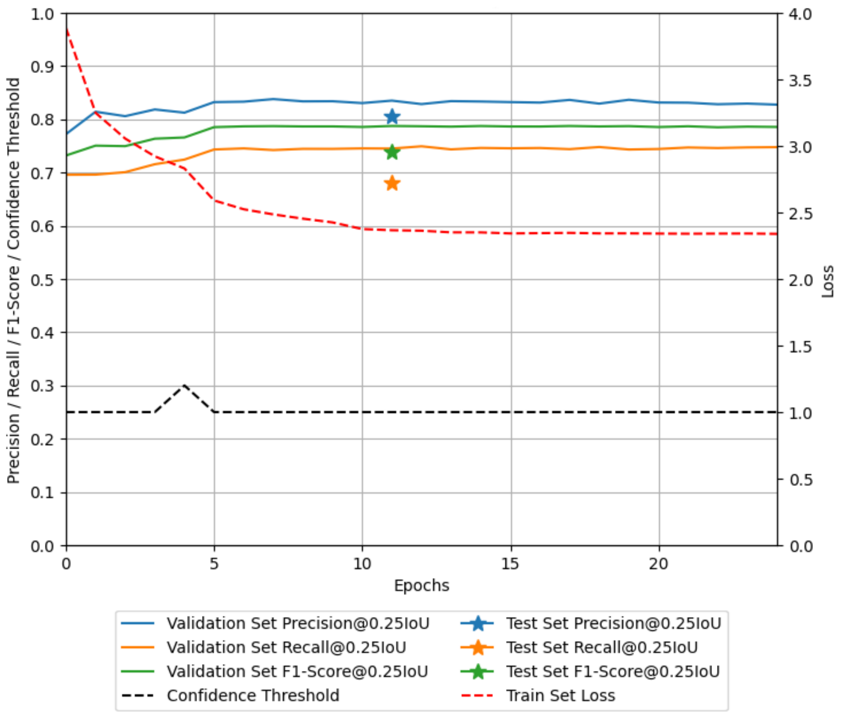


Figure A.1: Training convergence of SSD300 on KI-A.

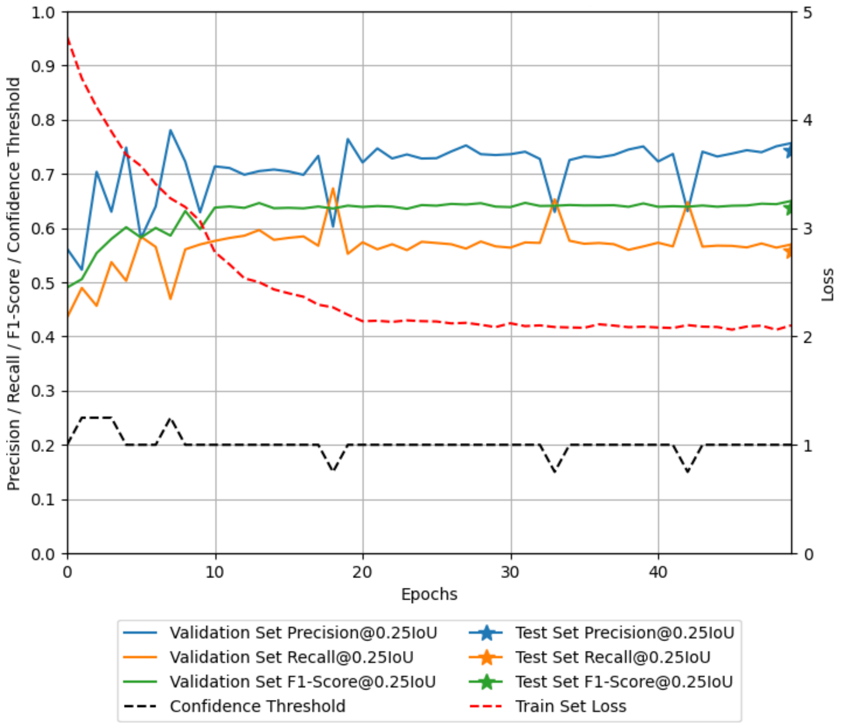


Figure A.2: Training convergence of SSD300 on CityPersons.

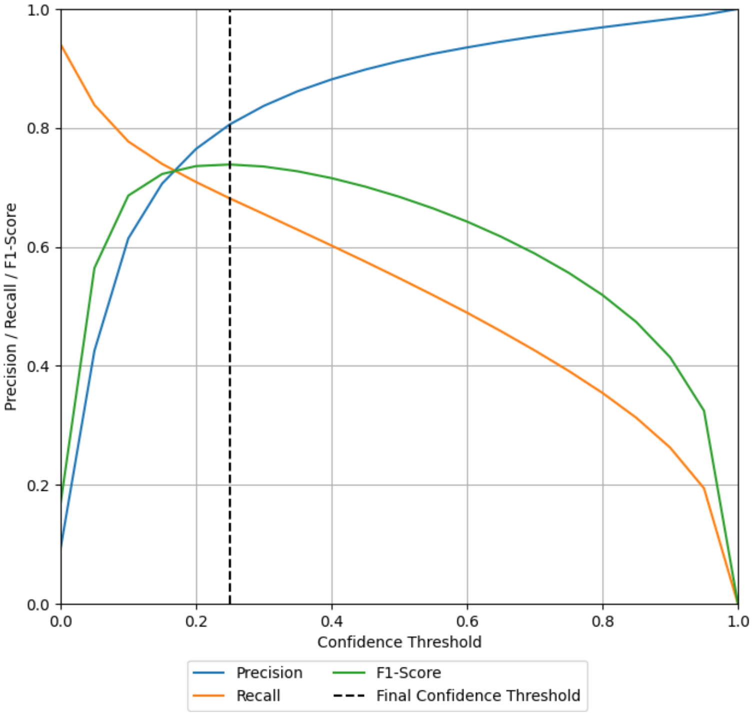


Figure A.3: PDSM over SSD300 confidence thresholds on KI-A.

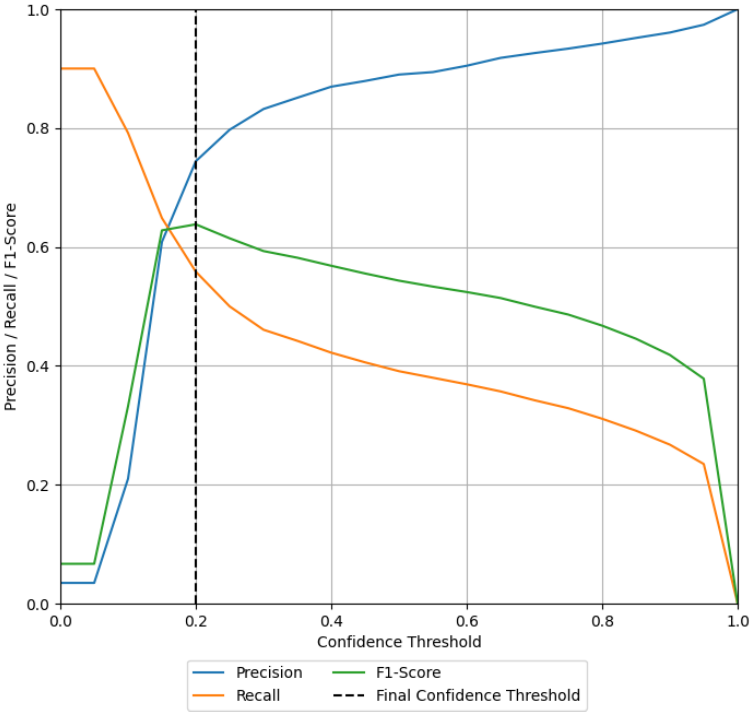


Figure A.4: PDSM over SSD300 confidence thresholds on CityPersons.

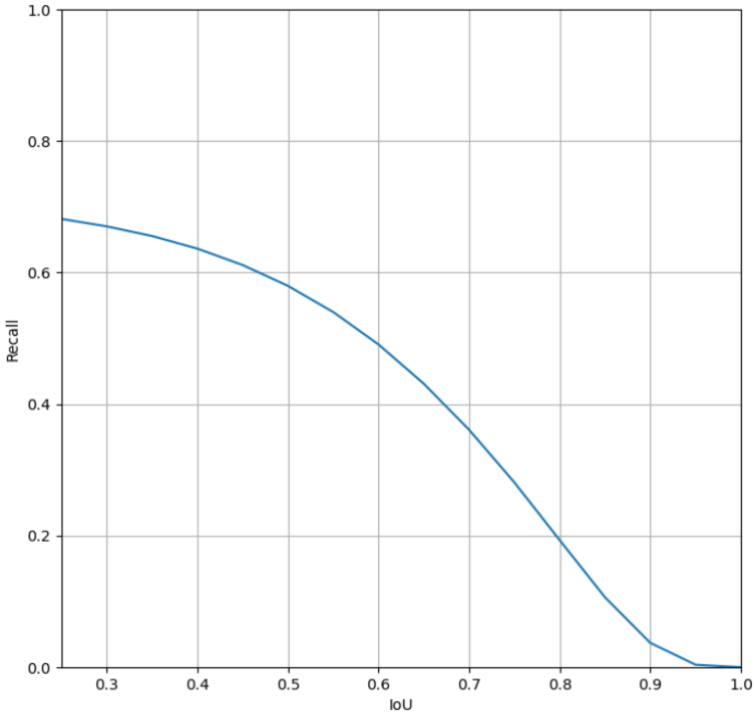


Figure A.5: Recall-IoU curve of SSD300 on KI-A.

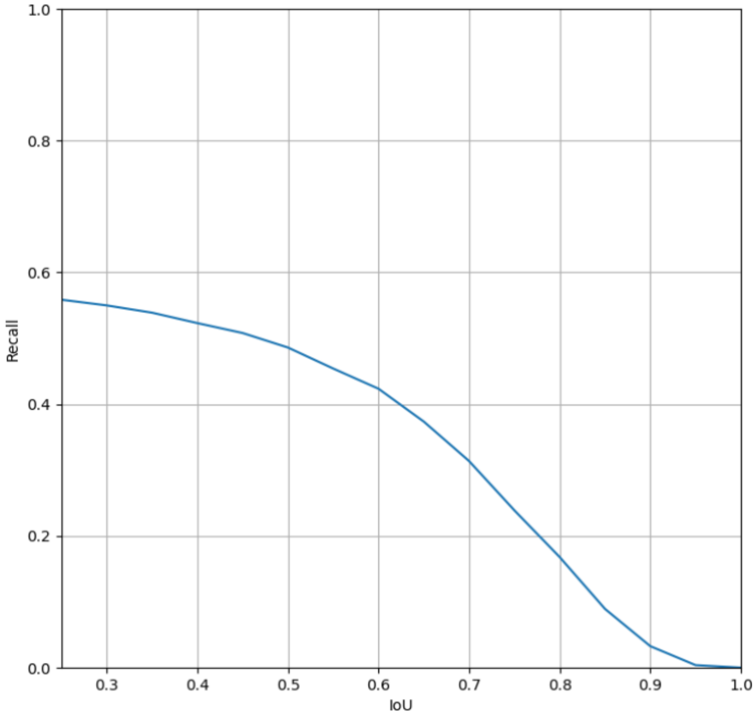


Figure A.6: Recall-IoU curve of SSD300 on CityPersons.

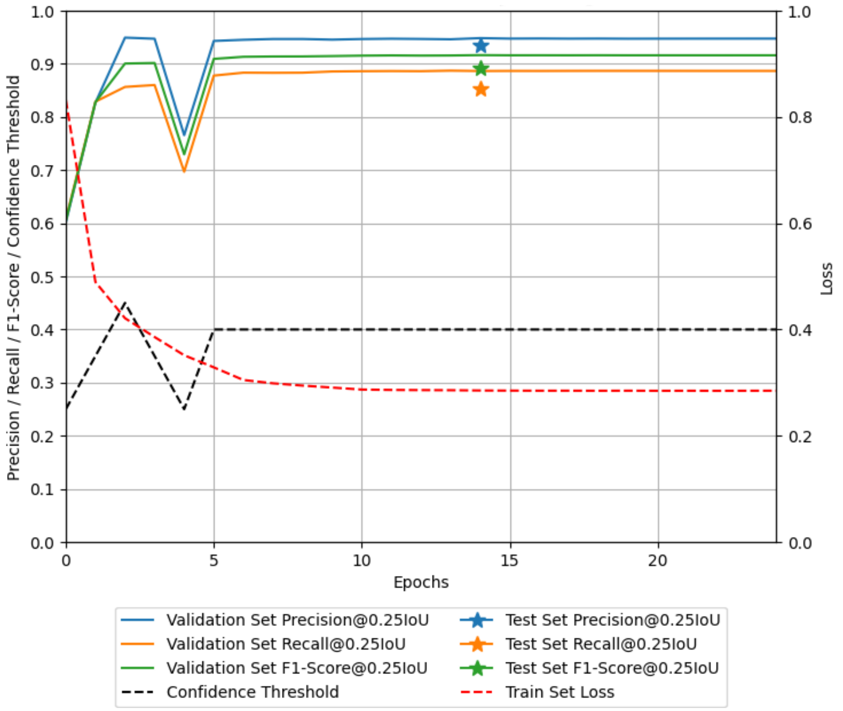


Figure A.7: Training convergence of RetinaNet on KI-A.

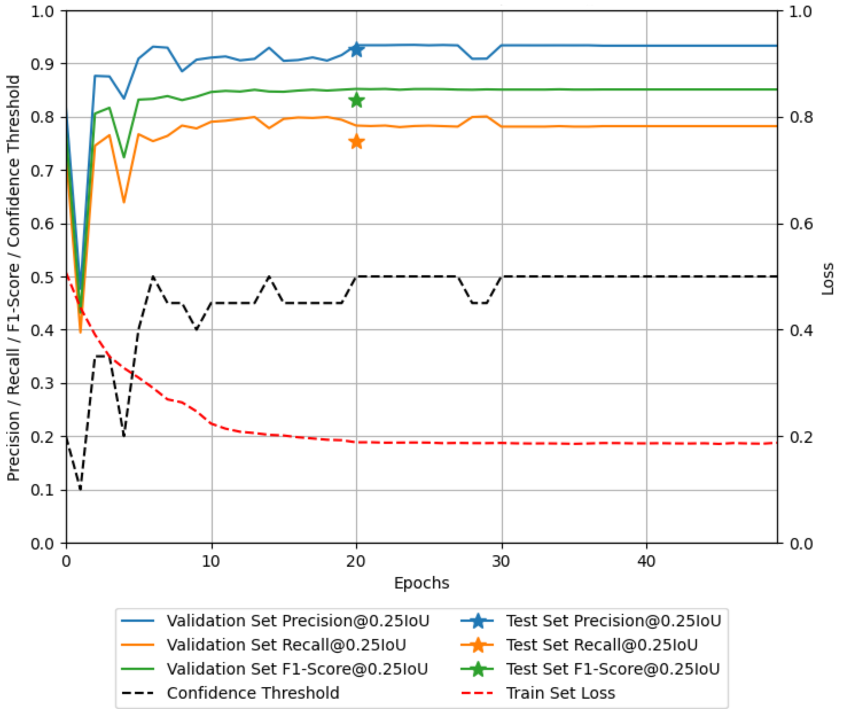


Figure A.8: Training convergence of RetinaNet on CityPersons.

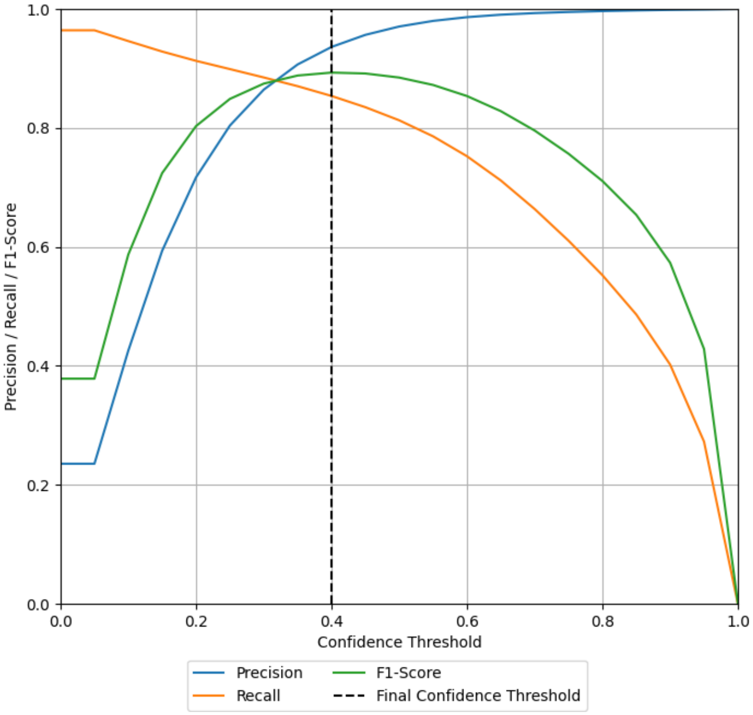


Figure A.9: PDSM over RetinaNet confidence thresholds on KI-A.

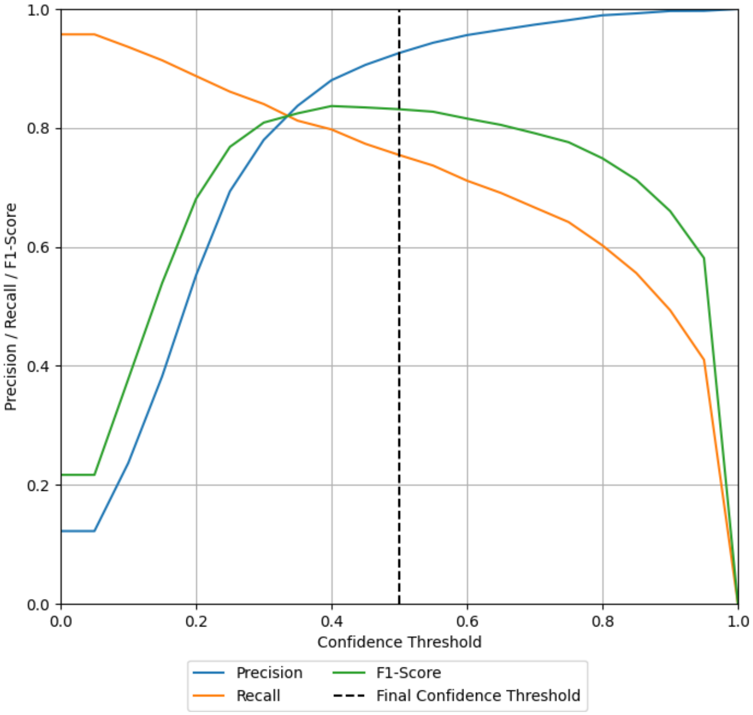


Figure A.10: PDSM over RetinaNet confidence thresholds on CityPersons.

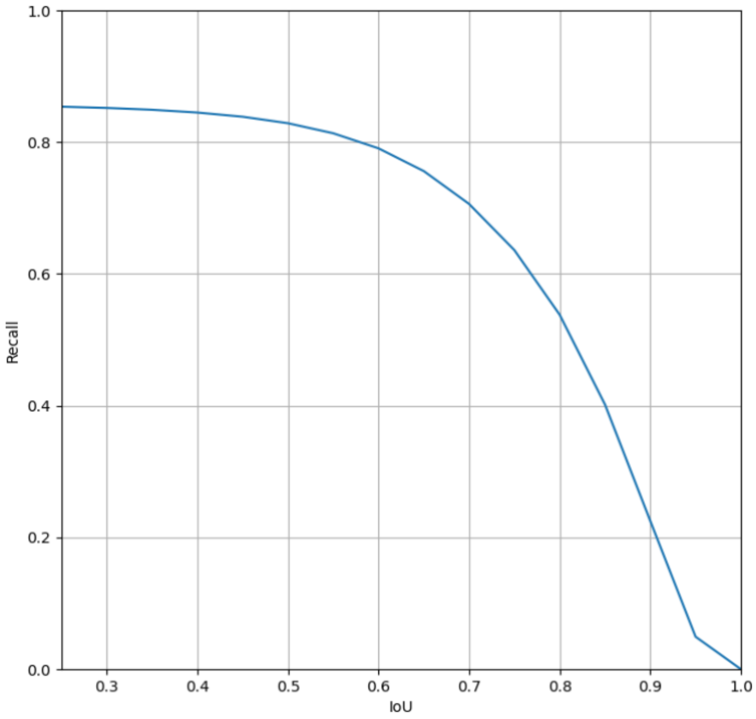


Figure A.11: Recall-IoU curve of RetinaNet on KI-A.

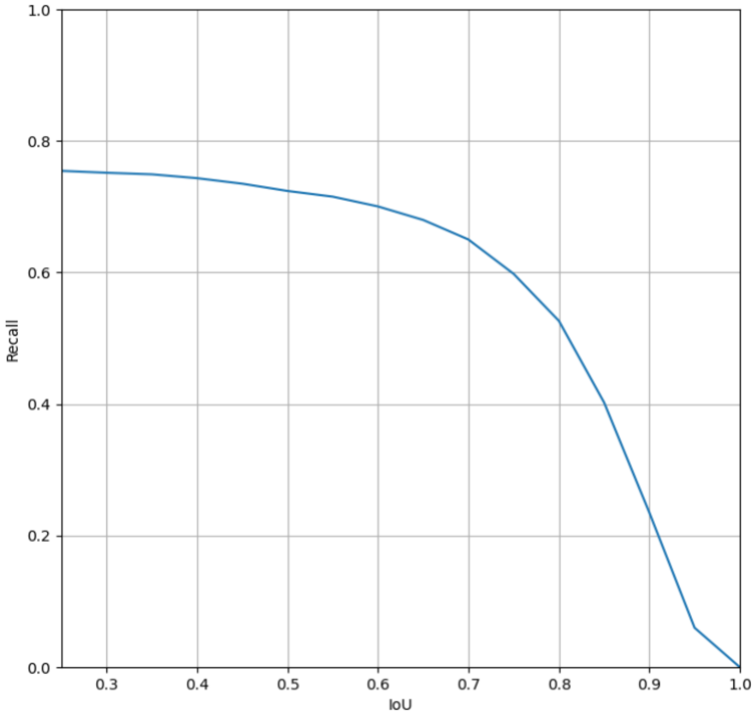


Figure A.12: Recall-IoU curve of RetinaNet on CityPersons.

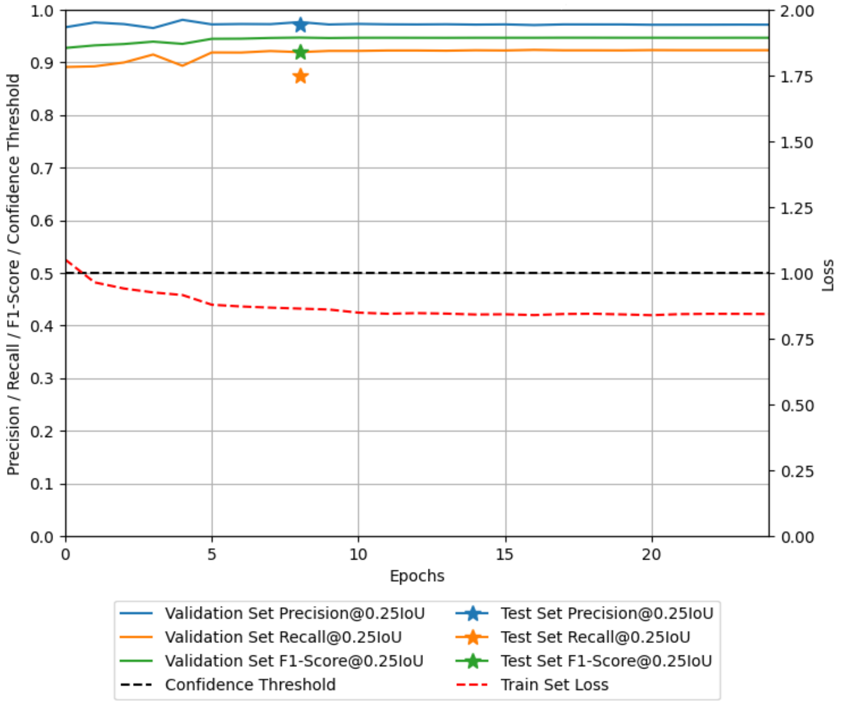


Figure A.13: Training convergence of FCOS on KI-A.

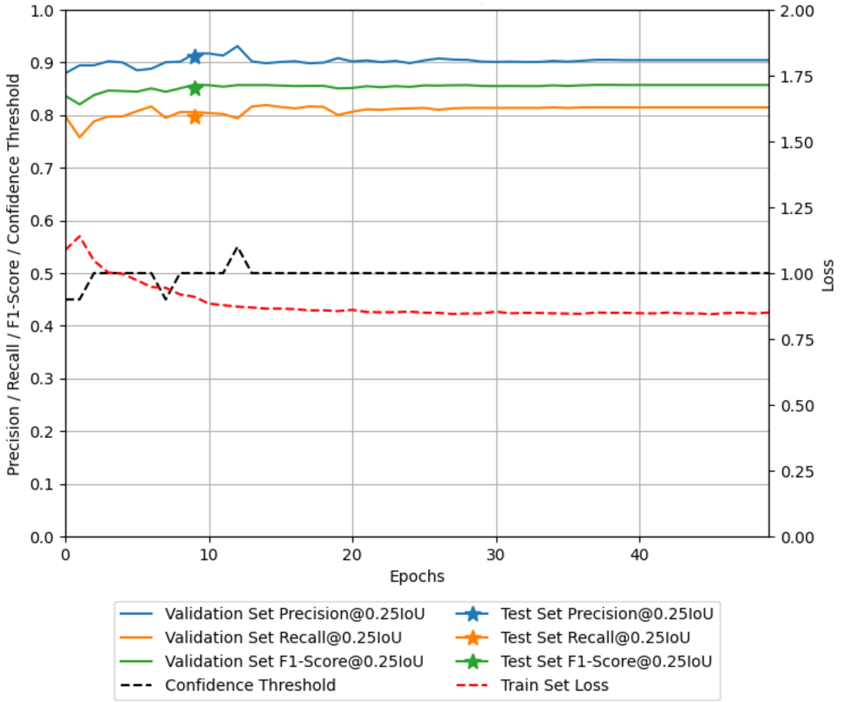


Figure A.14: Training convergence of FCOS on CityPersons.

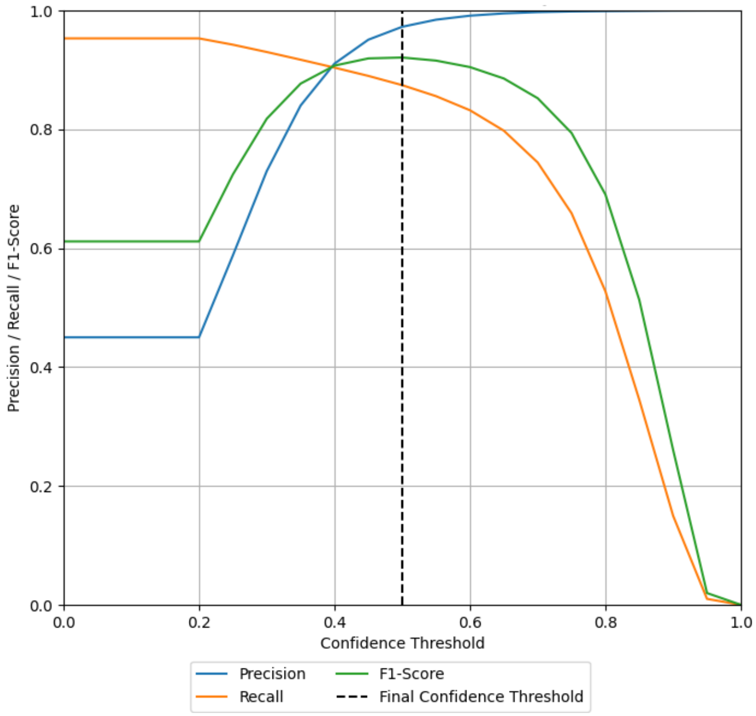


Figure A.15: PDSM over FCOS confidence thresholds on KI-A.

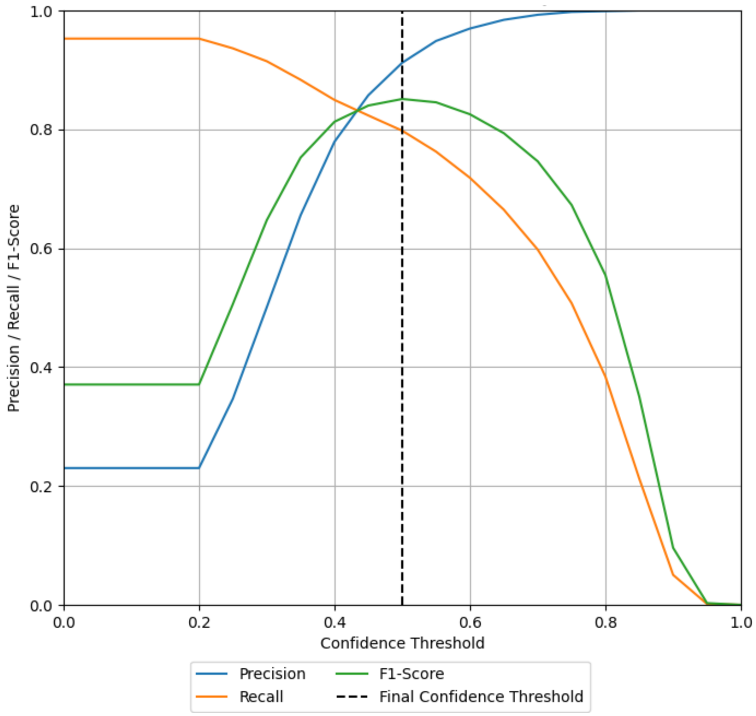


Figure A.16: PDSM over FCOS confidence thresholds on CityPersons.

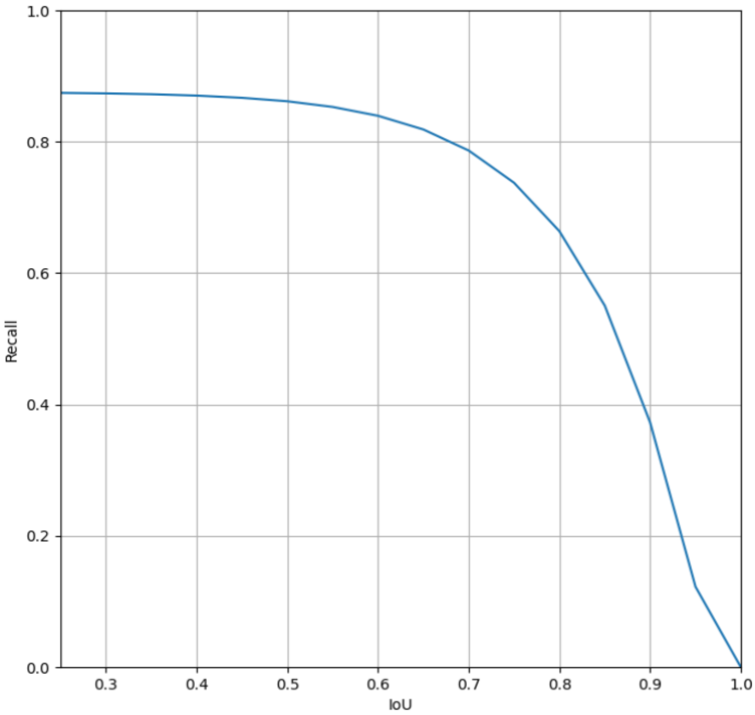


Figure A.17: Recall-IoU curve of FCOS on KI-A.

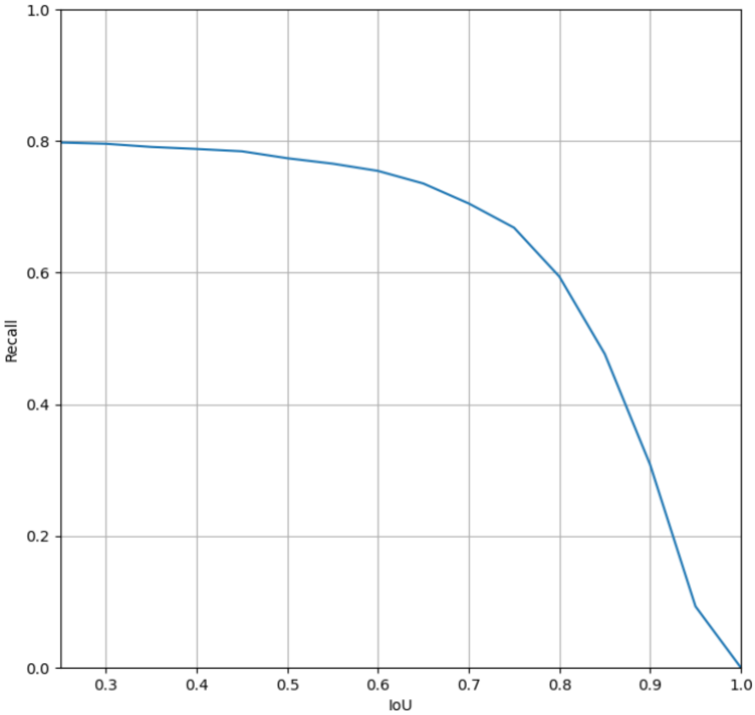


Figure A.18: Recall-IoU curve of FCOS on CityPersons.

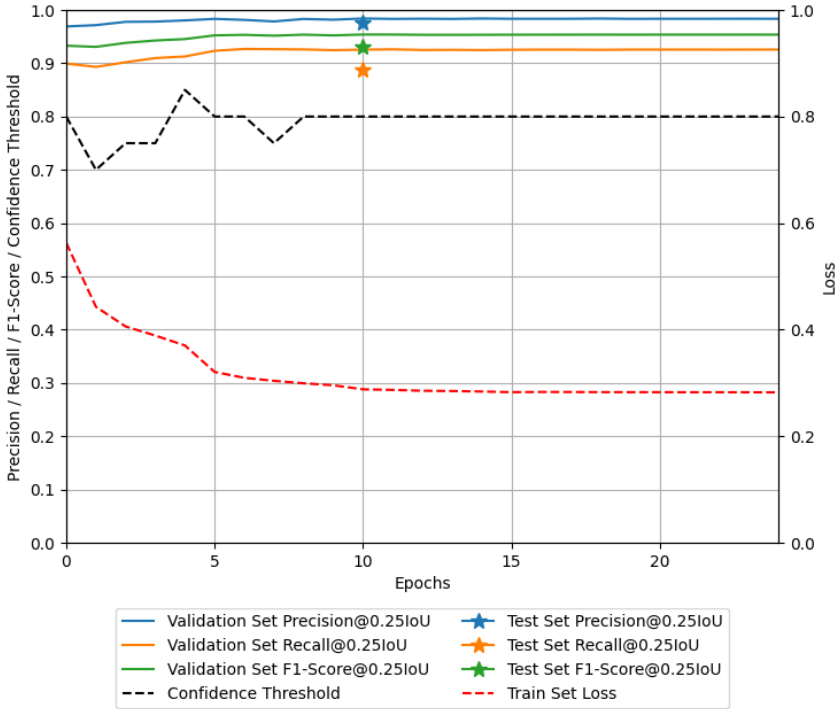


Figure A.19: Training convergence of Faster R-CNN on KI-A.

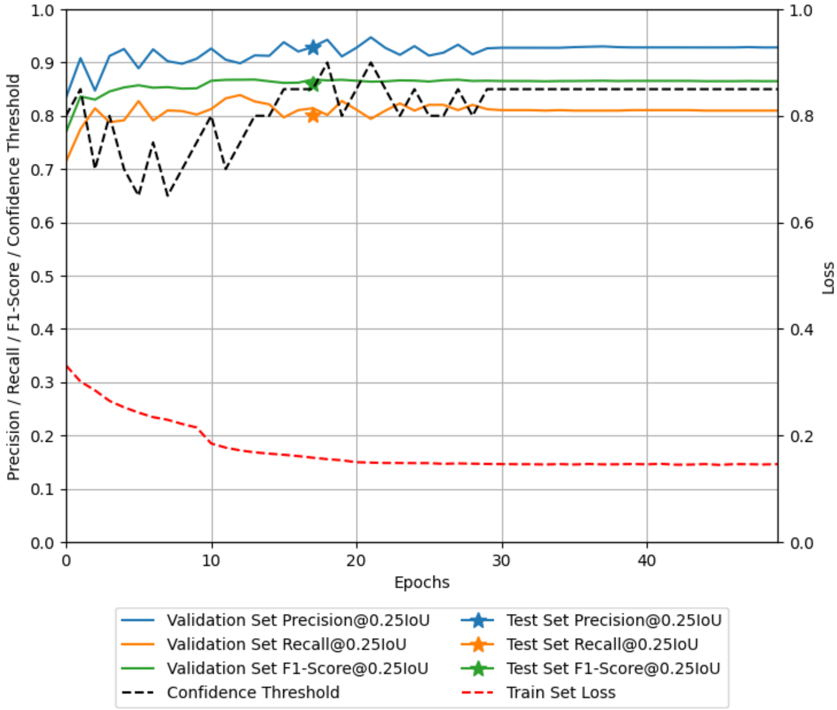


Figure A.20: Training convergence of Faster R-CNN on CityPersons.

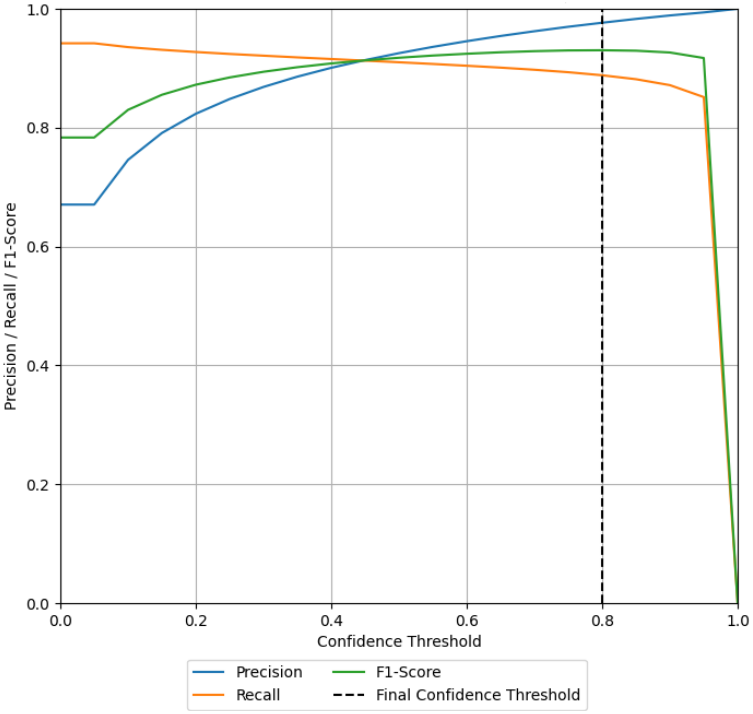


Figure A.21: PDSM over Faster R-CNN confidence thresholds on KI-A.

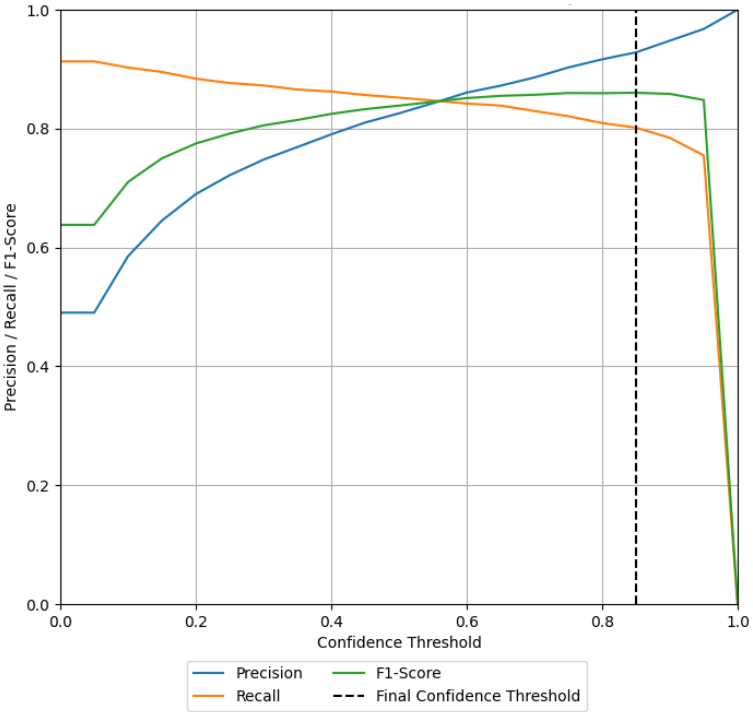


Figure A.22: PDSM over Faster R-CNN confidence thresholds on CityPersons.

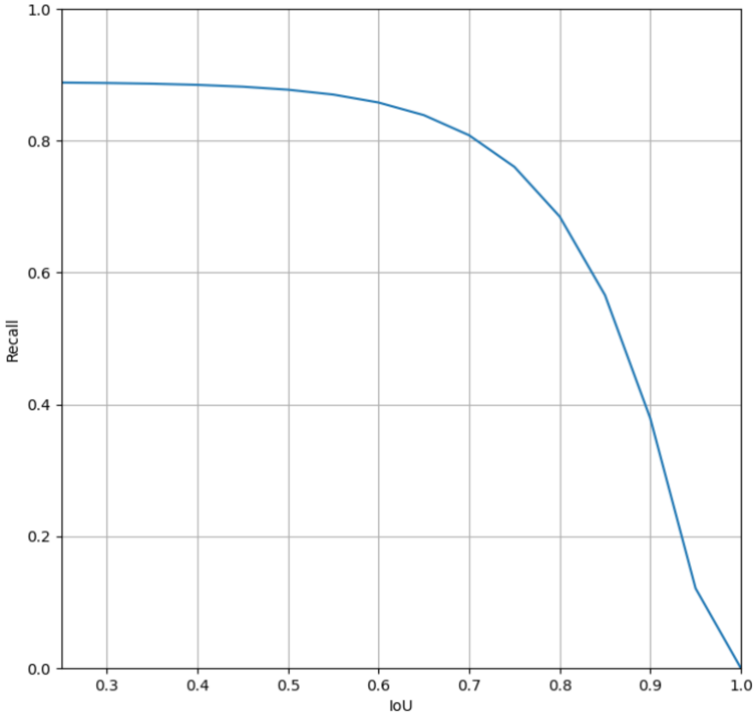


Figure A.23: Recall-IoU curve of Faster R-CNN on KI-A.

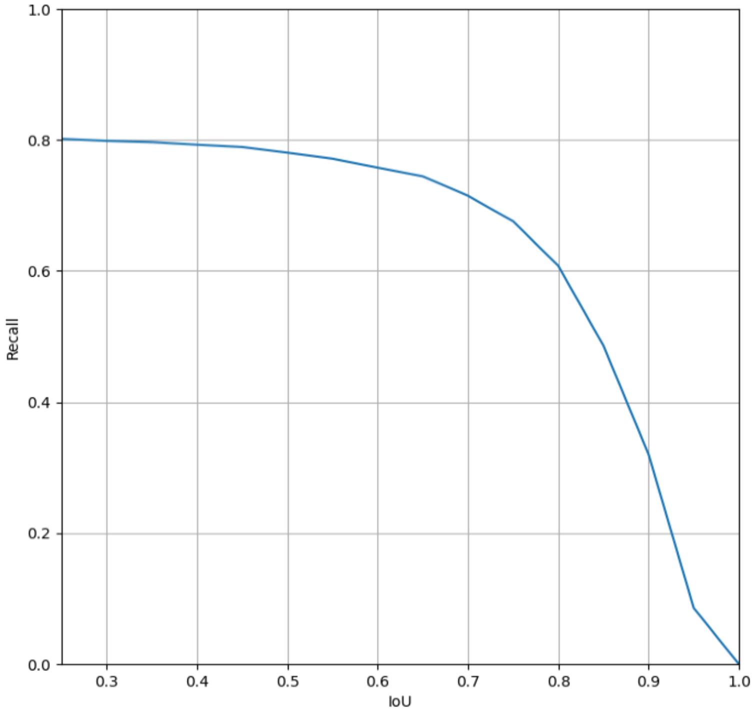


Figure A.24: Recall-IoU curve of Faster R-CNN on CityPersons.

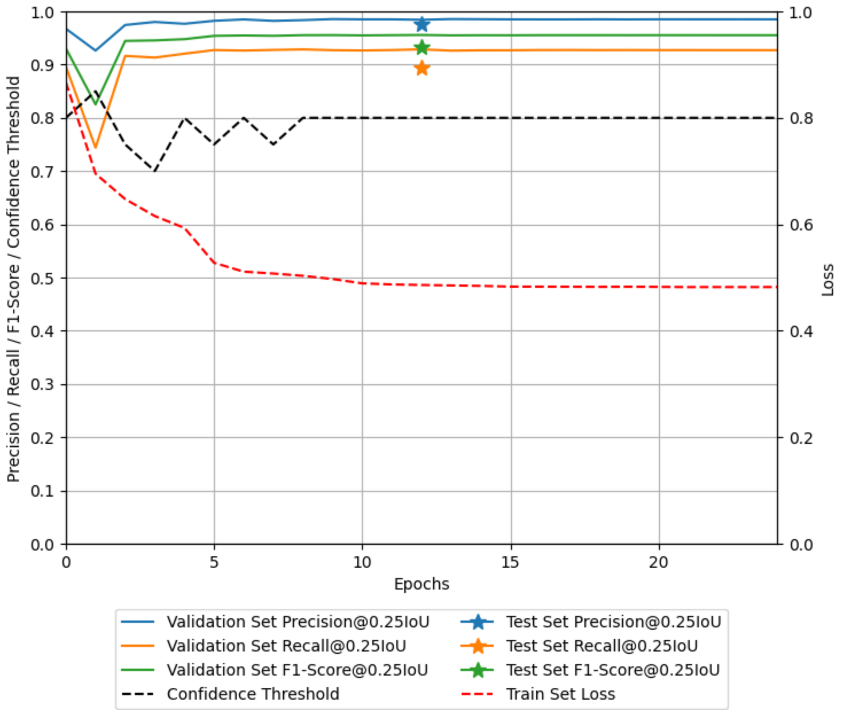


Figure A.25: Training convergence of Mask R-CNN on KI-A.

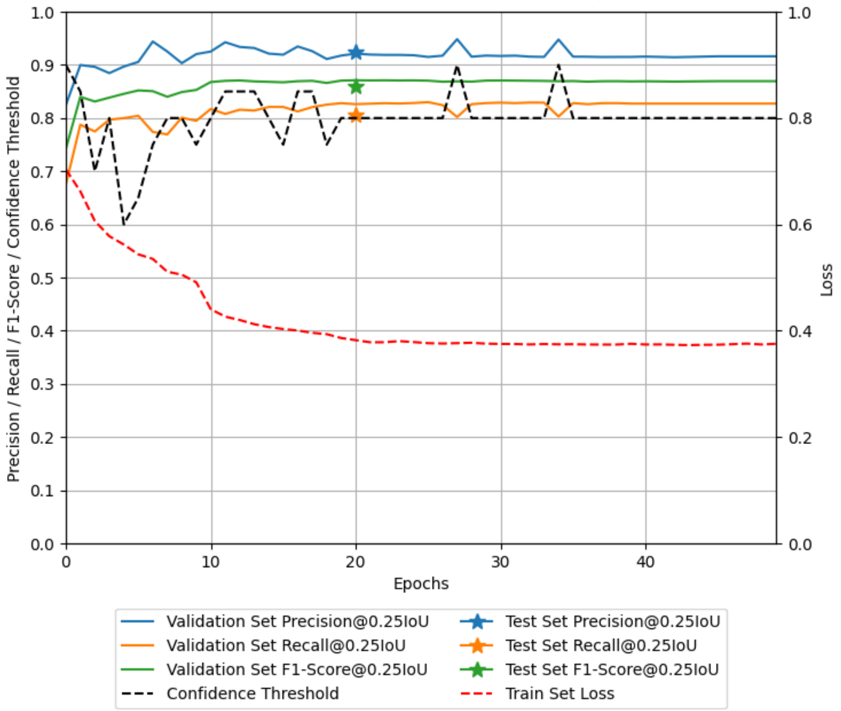


Figure A.26: Training convergence of Mask R-CNN on CityPersons.

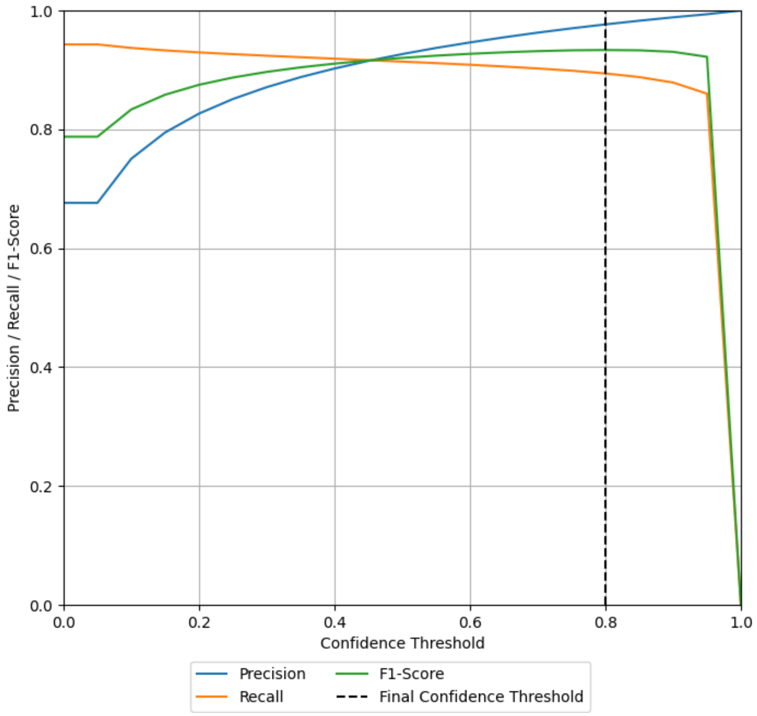


Figure A.27: PDSM over Mask R-CNN confidence thresholds on KI-A.

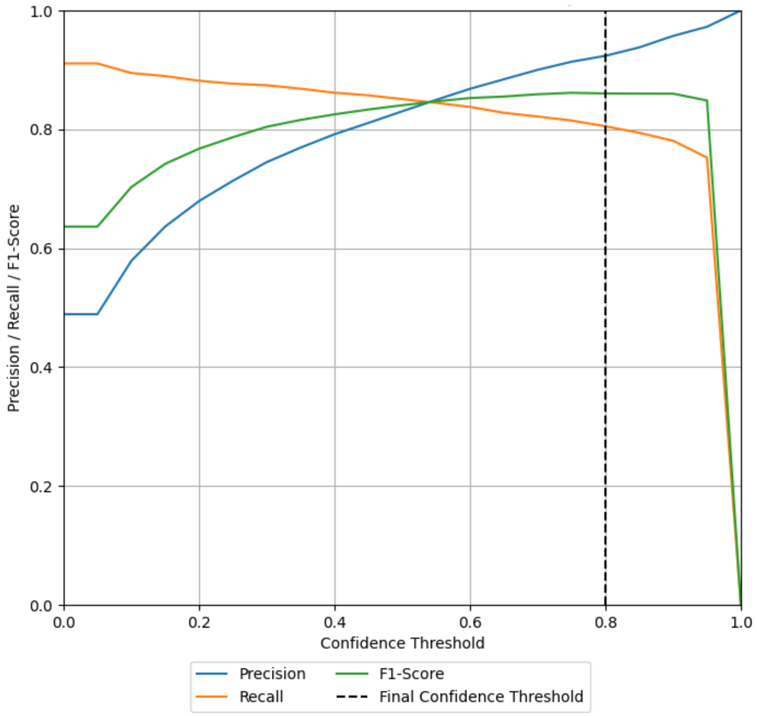


Figure A.28: PDSM over Mask R-CNN confidence thresholds on CityPersons.

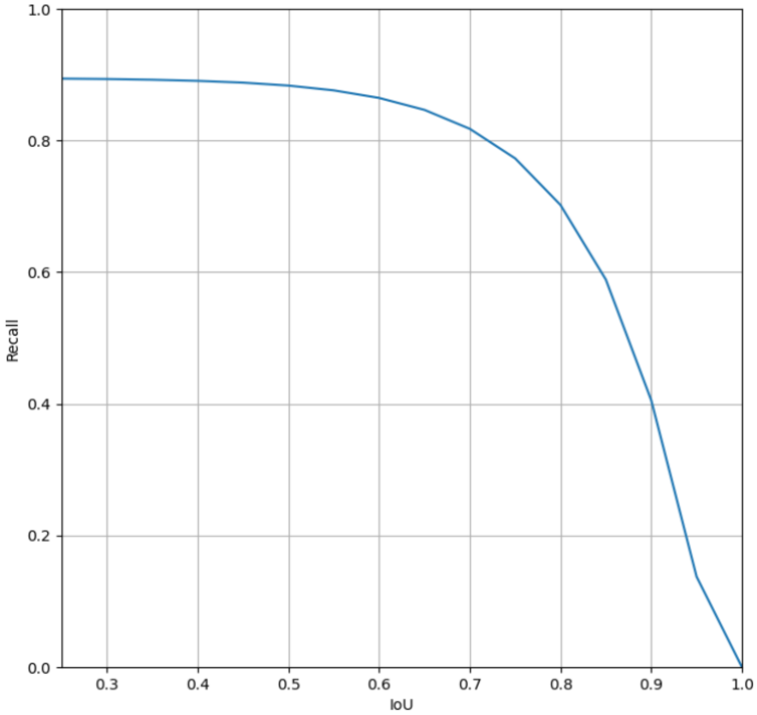


Figure A.29: Recall-IoU curve of Mask R-CNN on KI-A.

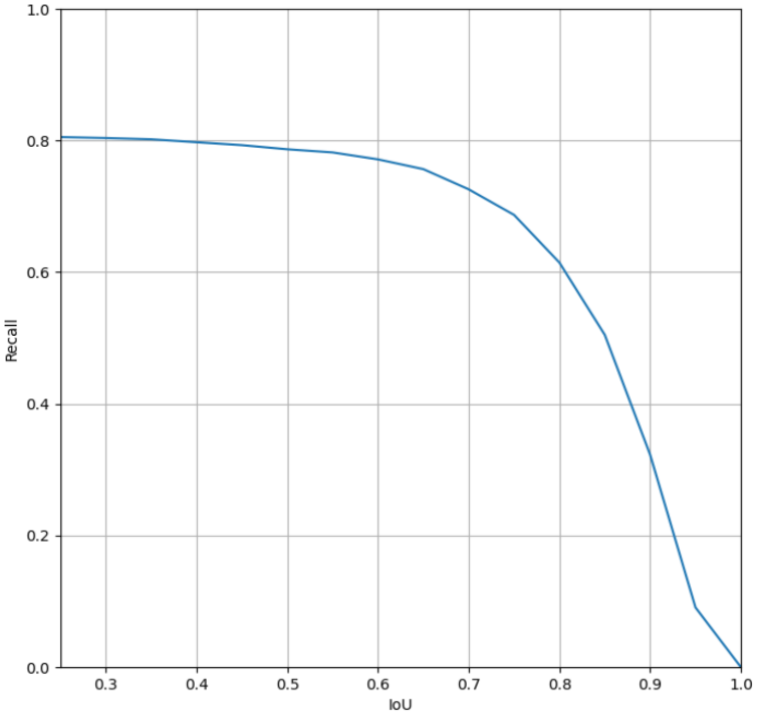


Figure A.30: Recall-IoU curve of Mask R-CNN on CityPersons.

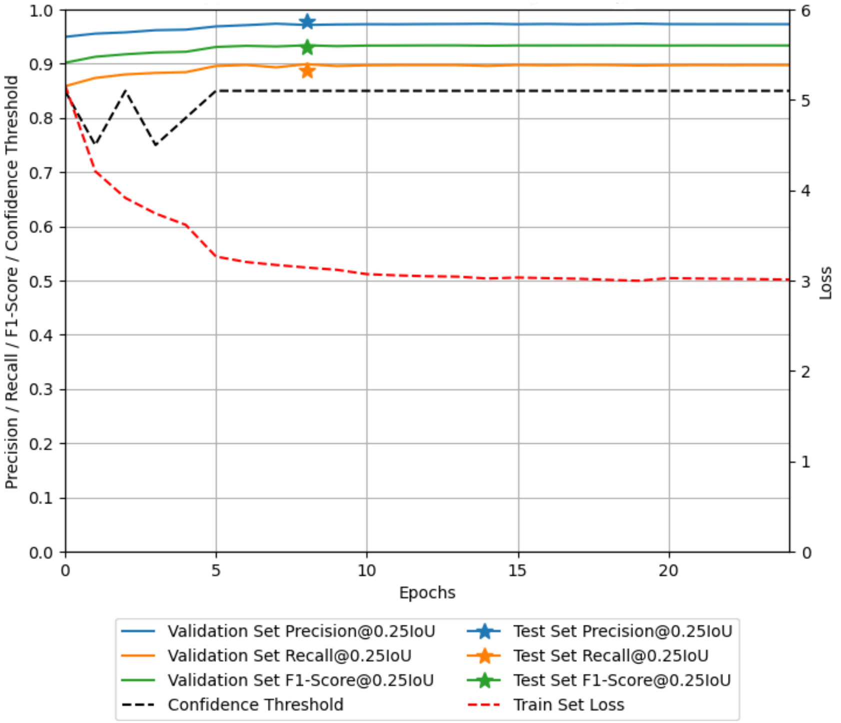


Figure A.31: Training convergence of Keypoint R-CNN on KI-A.

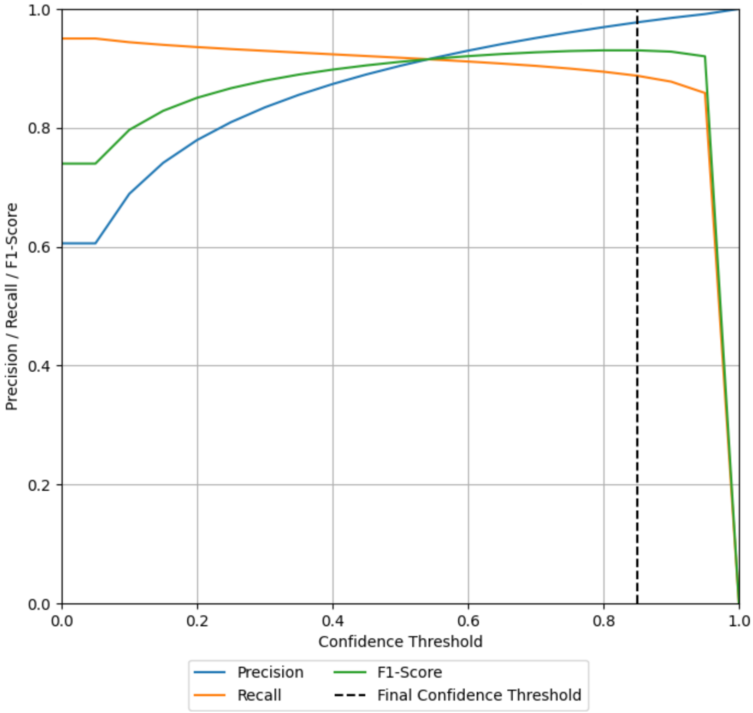


Figure A.32: PDSM over Keypoint R-CNN confidence thresholds on KI-A.

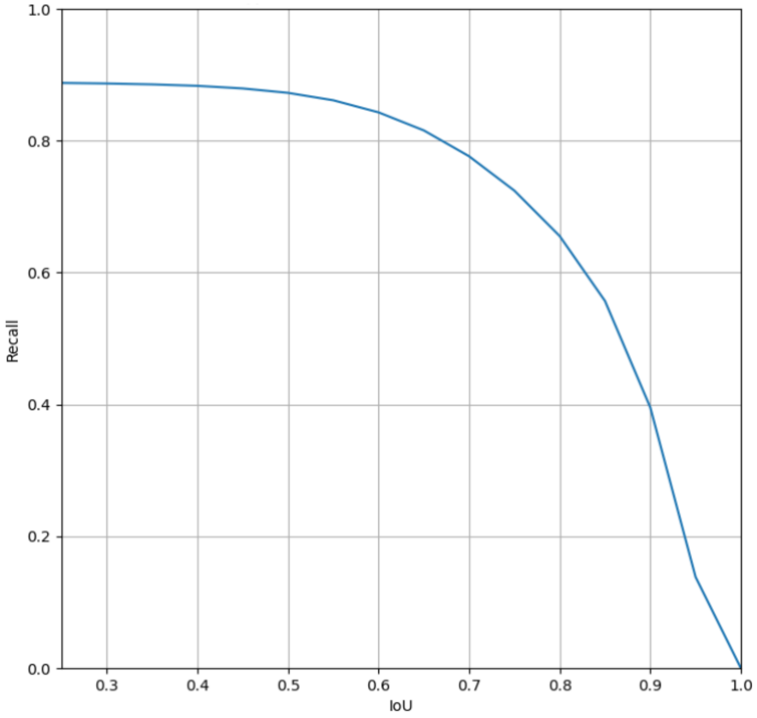


Figure A.33: Recall-IoU curve of Keypoint R-CNN on KI-A.

Appendix B

Supplementary Results of the PLF Analysis

This supplementary chapter contains further results related to the Performance Limiting Factor (PLF) analysis. These results include the correlation graphs for the **out of distribution** and **citypersons label** factors, which have not been considered PLF candidates for the main study.

For a factor to be considered a valid PLF candidate, it is required that the factor be well-defined and categorized based on its properties. Since the values of the **out of distribution** and **citypersons label** factors have been defined by the respective datasets, their natural categorization would be within the group of "meta annotations" factors. However, as "meta annotations" have been defined to represent environmental factors that are controlled by the KI-A simulation, the **out of distribution** and **citypersons label** factors have not been considered as part of the main study. Their effects on detection performance are reported in this supplementary chapter to provide additional insight into the DNN's behavior.

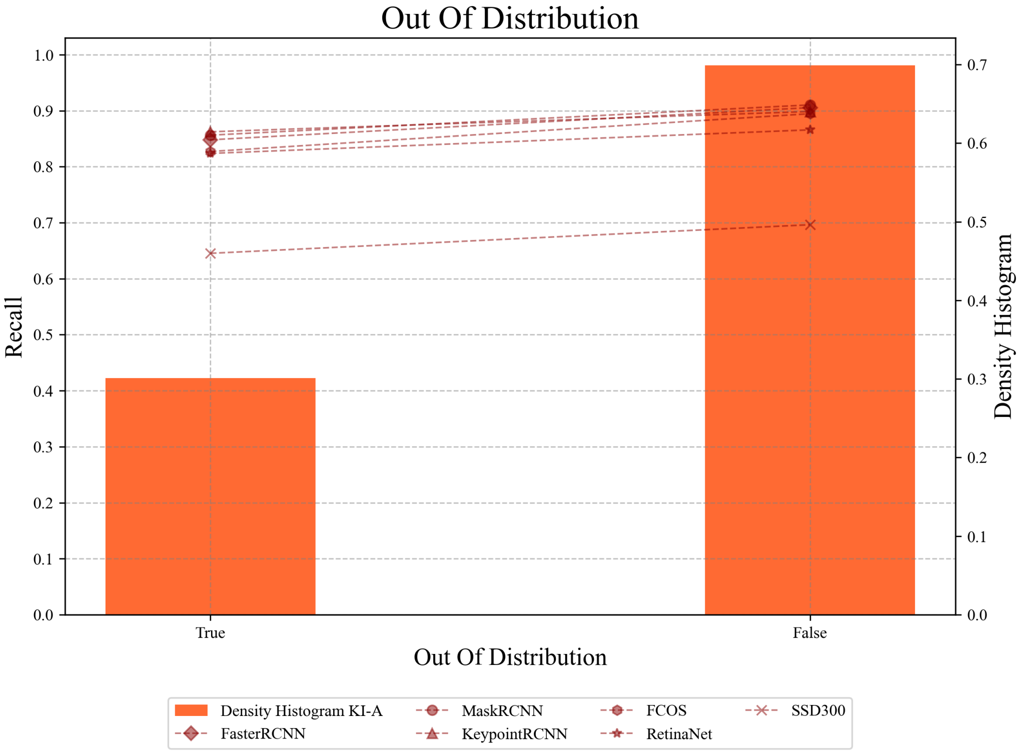


Figure B.1: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **out of distribution** factor.

The out of distribution factor is an object-based factor that is tracked by the KI-A simulation. Its values, represented by "True" or "False", indicate whether a pedestrian instance stems from a different batch of pedestrian instances that has not been shown to the pedestrian detectors during training, hence being marked as an out of distribution instance. The density histogram of this correlation graph represents the distribution of factor values within the final test split used for the evaluation. It can be observed from the distribution of factor values that roughly 30% of the split consisted of out of distribution pedestrian instances while the other 70% have been shown to the detectors during training. Furthermore, for pedestrian instances marked as out of distribution, there is a clear drop in detection performance to be observed; however, this performance decrease remains small and should be interpreted as proof of the detector's capabilities to generalize over unseen pedestrian instances.

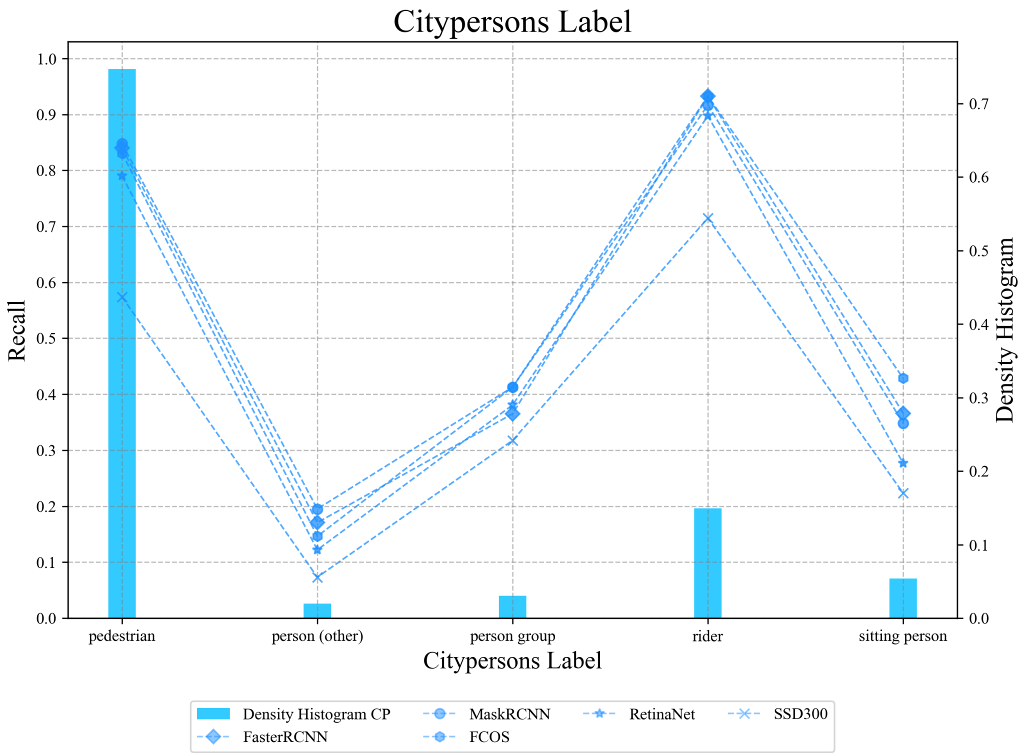


Figure B.2: Visualization of the correlation between the detection performance of the studied pedestrian detectors and the **citypersons label** factor.

The citypersons label is an object-based factor that represents the official class label of the CityPersons dataset. Even though all of the studied pedestrian detectors have been optimized for the detection of a single "human" class, there are still interesting effects to be observed from the correlation graphs of the respective citypersons class labels. The "pedestrian" class is most frequent within CityPersons, followed by the "rider" class, which represents cyclists and motorists. Interestingly, the detection performance for human instances marked as "riders" is higher compared to the ones marked as "pedestrian" even though the distribution of factor values is highly in favor of the "pedestrian" class. Furthermore, it can be observed that the detection performance on the remaining three classes, including "person (other)", "person group", and "sitting person" drops significantly, which indicates that the human pose represents a crucial factor that impacts the DNN detection performance.