Performance Limiting Factors of Deep Neural Networks for Pedestrian Detection

Yasin Bayzidi[1,2]
yasin.bayzidi@volkswagen.de
Alen Smajic[1]
alen.smajic@volkswagen.de
Jan David Schneider[1]
jan.david.schneider@volkswagen.de
Fabian Hüger[1]
fabian.hueger@cariad.technology
Ruby Moritz[1]
ruby.moritz@volkswagen.de
Alois Knoll[2]
https://www.ce.cit.tum.de/en/air/home/

[1] Volkswagen AG
Berliner Ring 2,
38440, Wolfsburg, Germany

[2] Technical University of Munich
Arcisstraße 21,
80333, Munich, Germany

Abstract

Deep Neural Networks (DNNs) for perception in automated driving have been extensively studied, while achieving strong results in detection performance on pre-annotated test sets. However, there has been a gap in the literature on a systematic analysis of DNNs behavior to investigate the factors contributing to their misbehavior. As part of DNNs safety, we propose to both analyze DNNs behavior in challenging scenarios as well as the respective factors that actually contribute to their misbehavior. Although some of such factors have been studied individually, there is not a thorough study to compare all together in a systematic manner to unveil the impact of each factor leading to DNNs failures. In this paper, we propose an approach to evaluate the DNNs performance limiting factors (PLF), and their contribution to the DNNs misbehavior. Accordingly, we analyze seventeen factors from the literature, introduce four novel factors and conduct an assessment on all of them to assess their potential as a PLF. Furthermore, we evaluate our results based on six state-of-the-art pedestrian detection DNNs including three detection tasks. For our experiments, we study a synthetic as well as a real-world dataset for pedestrian detection. We show that there exist various similarities and dissimilarities when comparing the PLF from a synthetic dataset to a real one, and discuss the causes and effects of such relations. Furthermore, we provide an approach to analyze the common factors from both real-world as well as synthetic datasets which might have similar effects on various DNNs performance.

1 Introduction

Deep Neural Networks (DNNs) are a static part of most of the visual perception systems within automated driving. There have been manifold publications introducing new archi-

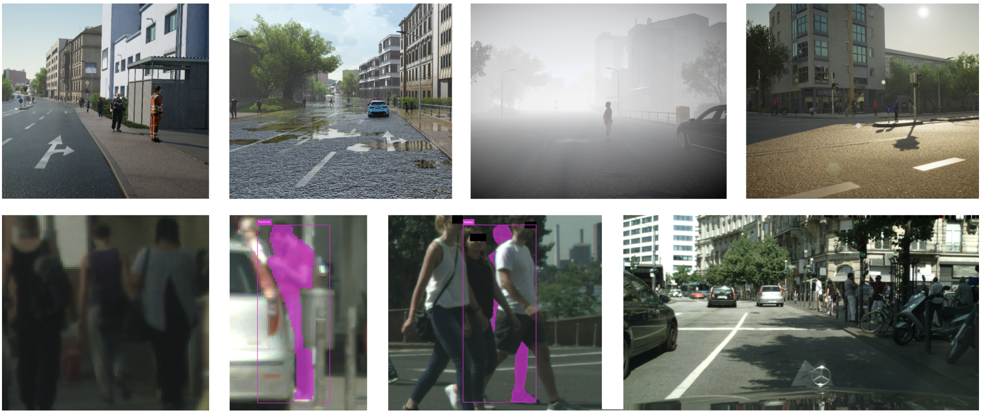

Figure 1: Examples of different factors contributing to the possible performance drop of the DNNs. The top row examples are from the KI-Absicherung dataset which include distance, wetness, fog and contrast factors. The bottom row examples are from the CityPersons dataset, which include brightness, occlusion, crowdedness and distance factors.

tectures, training methods, loss functions, *etc*. to increase the detection performance of such DNNs on one hand, and their efficiency in training as well as inference on the other hand [2, 16, 17, 39]. However, like any other data driven method, DNNs are also affected by the quality of the dataset used for their training. Therefore, it is important to focus not only on *e.g*. the architecture, and training procedure but also the dataset used for training. Based on that, there have been previous publications focusing on the factors present in the dataset that might contribute to the overall DNNs performance [11, 23, 32, 34, 38]. In other words, factors, whose existence or their lack of existence can be crucial for the DNNs to provide an accurate prediction. However, to the best of our knowledge, no previous publication has reported a systematic performance limiting factors analysis on DNNs that can lead to comparable conclusions about the impacts of such limiting factors on the DNNs performance.

In this paper, we discuss the conditions that define a performance limiting factor (PLF) and the properties that one needs to consider while studying PLFs. Based on that, we study the correlation of 21 factors and discuss their effect as a PLF individually, out of which, four are novel factors that are introduced in this paper. Furthermore, we perform our large scale analysis on six of the state-of-the-art (SOTA) DNNs that cover three detection tasks including 2D object detection, 2D instance segmentation, and keypoint detection. Based on that, our contributions are as follows:

- We discuss the conditions that would define a performance limiting factor.
- We collect and analyze 17 factors from the literature.
- We extend the known limiting factors to 21 by introducing four novel factors.
- We perform cross model as well as cross task analysis including six DNNs and three detection tasks on a synthetic and a real-world pedestrian detection dataset.

The rest of the paper is organized as follows: we introduce the related work from the literature in Section 2; the definition of the PLF as well as all the considered factors are introduced in Section 3; the experiment setup is discussed in Section 4; the results are reported and discussed in Section 5; finally, the paper is concluded in Section 6.

2 Related Work

The recent works in tackling the causes behind the DNNs miss-performance approach this problem with different perspectives, from deep analysis of the training dataset to find causes of failures to using such causes against the DNNs for adversarial attacks. Gauerhof *et al.* [11] introduced various image artefacts which affect the performance of a DNNs and thus the overall safety of the system. Saad and Schneider [34] studied the effect of image vignetting on the DNNs training, while Tian *et al.* [37] used it as an adversarial attack. Lens flare is another camera artefact added to the images which can cause significant performance drop in DNN [6]. Therefore, Wu *et al.* [41] introduced methods on how to train DNNs on images with flare, Nussberger *et al.* [26] discussed the ways to do object tracking in such cases, and Qiao *et al.* [30] introduced a method to remove lens flare from the training images.

Ramanagopal *et al.* [32] reported that the DNNs perform better in sunny weather than in cloudy and general overcast weather. Teeti *et al.* [36] and Qu *et al.* [31] used CycleGANs to improve the detection performance in night time, low illumination and the cases of water droplets. Sotelo *et al.* [35] also reported that sunrise and sunset and different wetness levels can affect the performance of a DNN. Yoneda *et al.* [42] and Michaelis *et al.* [24] discussed the challenges for identifying adverse weather conditions such as sun glare, rain, fog, and snow. Bijelic *et al.* [4] introduced a new dataset that includes multi-sensor data and covers different weather conditions. Ogunrinde and Bernadin [27] used a CycleGAN to defog the images to increase the performance, while Huang *et al.* [15] introduced the DSNet, which is an architecture designed to tackle object detection in fog. Kenk *et al.* [18] introduced the DAWN dataset, which contains different weather conditions including fog, snow, rain and sandstorm. Musat *et al.* [25] and Von Bernuth *et al.* [3] introduced methods to augment different weather conditions to the image samples, which led to a boost in the DNNs detection performance. Pepe *et al.* [29] had used CNN and bi-directional long-short term memory networks (BLSTM) to detect the wetness type of the road by using the noise coming from the tires recorded by specific microphones.

The effect of scenes crowdedness and the relative positions of the objects on the DNNs performance are also studied. Based on that, Wang *et al.* [40] and Liu [21] reported on the challenges of detecting pedestrians in the crowd and suggested different methods for increasing the detection performance in such cases. Besides that, Lyssenko *et al.* [23] reported on the effect of distance on the pedestrian detection DNN. Moreover, Bayzidi *et al.* [2] implemented a pipeline for augmenting different occlusion patterns into images to study the effect of occlusion in classification tasks. To the best of our knowledge, we could not find sophisticated test based approaches to study the above mentioned cases in a single setup to unveil the effect of such challenges compared to each other. Therefore, we implement and test all of the above mentioned factors and introduce four novel factors. We show how one can use such a conducted evaluation to shortlist the factors that have high impact in detection performance which might lead to developing strategies to increase the detection performance based upon.

3 Method

In this section, we first discuss the definition of a performance limiting factor (PLF) and then introduce all the factors that are implemented and tested in detail.

Performance Limiting Factor (PLF): A factor is considered performance limiting, if the presence of the respective factor in the input data causes significant drop in detection accuracy of the DNNs, such as precision and f1 score for image level and recall for object level factors. An intuitive way of considering a factor as PLF would be to analyze the correlation of such a factor to the DNN's performance. However, one should notice that the performance of a DNNs might be low in specific PLF levels due the under-representation of the data in such levels, that would cause the DNNs not to converge well for such data, or over-fit to other levels that are more frequent in the training dataset. Therefore, we consider a factor as a PLF, if there exists a correlation of such a factor with the performance of the DNNs regardless of the frequency of such a factor in the training dataset. In the following, we introduce the factors we have analyzed to be considered as a PLF. These factors are categorized into three categories based on the properties they are addressing, which include image intensity, geometrical properties, and meta annotations. Moreover, they can either be extracted per object sample or image sample.

3.1 Image Intensity

The factors falling into this category are the ones that are extracted based on the image pixel intensity values, which often require image processing methods to extract. Therefore, we utilized established image processing techniques to extract them.

Edge Strength: The edge strength $s(X)$ represents the magnitude M of the edges presented in the image X, which is converted to grayscale in advance. The magnitude is calculated as follows:

$$M(X) = \sqrt{dx(X)^2 + dy(X)^2}, \tag{1}$$

while dx and dy are calculated using a Sobel filter [5] to extract the vertical as well as the horizontal edges presented in the image. As the magnitude function returns a vectors of size $x \times y$, where x and y are the width and height of the input image respectively, we convert this vector to a single normalized number to be quantified as follows:

$$s(X) = \frac{1}{P(X) \times 255} \sum_{i}^{P(X)} M(i), \tag{2}$$

where $P(X)$ represents the number of pixels of the input image X.

Boundary Edge Strength (ours): Indicates cases when the objects boundary is blended into the surrounding background due to low illumination or color similarity, *e.g.* when a pedestrian is wearing a black jacket and standing in front of a building with a dark texture. To calculate this factor, we simply take the magnitude vector $M(X)$ calculated in (1) and mask it based on a boundary mask, which we compute by subtracting a dilated version of the binary pedestrian mask from an eroded version to extract a zone around the boundary of the object. We then take the mean of this masked vector based on (2).

Background Edge Strength (ours): Indicates cases with highly salient features in the background that might affect the decision of DNNs by deviating its attention to such highly salient features. Therefore, we mask the magnitude vector $M(X)$ from (1) after converting the image to greayscale to remove the pedestrian from it and take the normalized mean of the remaining area using (2).

Contrast: This factors indicates the contrast of the image as a limiting factor. The contrast of a grayscale image is calculated as follows:

$$c = \sigma(X^{'})/128, \tag{3}$$

where σ is the standard deviation of the input vector, and $X^{'}$ is the masked input image to contain only the object bounding box vector.

Contrast to Background (ours): Indicates the contrast difference between foreground, i.e. the object, and its surrounding background for the cases when there is a low contrast object in high contrast background, *e.g.* in a sunrise time, when the camera is facing towards the sun, the object in the foreground might have a very low contrast, as the camera lenses are capturing the strong light beams from the sun. To do so, the contrast of the foreground and background are calculated using (3). Afterwards, the absolute difference of the two contrast numbers is taken as the value for this factor.

Brightness: The simple mean of the grayscale image vector normalized by diving by 255 is calculated as the brightness factor.

Foreground Brightness: The mean of the image intensities inside the bounding box area is calculated as the foreground brightness.

Object Entropy: The Shannon Entropy of the area within the bounding box is extracted for this factor. Based on that, objects which have a higher entropy, tend to contain more information, and the ones with lower entropy would have less information within them, therefore causing the DNNs to struggle in detecting them. To calculate the Shannon Entropy of an object, we used the Shannon Entropy formula in [12] directly from the Scikit-Learn library [28] as follows:

$$H(X) = \mathbb{E}_{X\sim P}[I(X)] = -\mathbb{E}_{X\sim P}[logP(x)], \tag{4}$$

where X is the input image cropped to the bounding box of the studied object, $I(X) = logP(X)$ represents the information content of X, and $P(X)$ is the number of pixels in in X.

3.2 Geometrical Properties

These factors are extracted from the geometrical properties of the object, out of which, some might require different annotation formats, which are explained in the following.

Crowdedness (ours): We define crowdedness as a quantitative value calculated per object which indicates if an object is close to other objects of the same class. To calculate this value, we take the cumulative overlap of its bounding box b with the nearby bounding boxes of the same class $b^{'}$ as follows:

$$O(b,b^{'}) = \frac{\min(A(b),A(b^{'}))}{\max(A(b),A(b^{'}))} \times |b \bigcap b^{'}|, \tag{5}$$

where $A(b)$ is the area of the bounding box b and $|b \bigcap b^{'}|$ is the intersection of the two bounding boxes. We then calculate the crowdedness factor $C(b)$ as the cumulative relative overlap of one bounding box with all the other bounding boxes in the same input image as follows:

$$C(b) = \sum \frac{O(b,b^{'})}{A(b)} \qquad \forall b^{'} \in \mathbb{B} \setminus b, \tag{6}$$

where $\mathbb{B} \setminus b$ is the set of all the bounding boxes except the current one in the input image.

Bounding Box Height: Bounding box height is defined as the height of the object in terms of number of pixels based on the bounding box dimensions.

Bounding Box Aspect Ratio: Indicates the cases when the bounding box is distorted because of uncommon posture of the object or the view angle of the camera, which is calculated as the height to width ratio.

Visible Instance Pixels: There might be cases, when the bounding box has a normal aspect ratio, but the number of pixels which are truly representing the studied object are changing. This factor can reveal such cases when studied along with other factors such as bounding box height. However, to extract the visible pixels, one needs to have the semantic instance segmentation annotations, which is not available in many datasets.

3.3 Meta Annotations

The factors of this category cannot be extracted and are annotated as meta annotations either for the whole image or for the object. Therefore, this depends on whether the dataset contains these meta annotations.

Lens Flare Intensity: There are cases when the camera is facing towards a light source such as sun, which might introduce an artefact called lens flare. The lens flare intensity is not easily measurable in a real dataset. Therefore, we analyse this factor in our experiments based on the synthetic dataset we used for training. The data samples with lens flare are generated directly by the simulation engine under certain environmental conditions, such as the camera facing sun with specific angles.

Vignette Intensity: Vignette is a filter that is applied to the images to darken the edges around the image. The aforementioned synthetic dataset also includes samples with different vignette intensities along with their intensity values.

Fog Intensity: Depending on the humidity and other weather conditions, fog can be present in the input data with different intensities, which are not easily quantifiable in real world. Similar to the previous two factors, this factor is also directly available in the used synthetic dataset, which indicates the intensity of the fog added to the individual image samples.

Daytime Type: The daytime type is a categorical factor which includes three different categories including “day”, “low sun position” and “medium sun position”.

Sky Type: Sky type is a categorical factor including “clear”, “partially clouded” and “fully clouded”.

Wetness Type: As another categorical factor, this factor represents different weather conditions that can cause different wetness levels on the streets. This factor includes the values “dry”, “slightly moist” and “wet with puddles”.

Occlusion Ratio: This factor represents the relative occlusion ratio, i.e. the amount of visible area compared to the object full area. However, estimating the occlusion can not be easily done, when the full object segmentation masks are not available. Fortunately, we were able to estimate this factor also for the CityPersons [43] real dataset using the instance segmentation masks provided by the dataset.

Truncated: Similar to occlusion, this factor indicates if the object is truncated by being on the edges of the input images and therefore cropped to the image dimensions. However, it is only a binary factor representing if the object is truncated or not, which can only be considered if the annotations provided by the dataset include such information.

Distance: Indicates the distance of the object to the recording camera in meters. This can be extracted from the LIDAR information, which was provided for both of the studied datasets.

	Task	CityPersons			KI-Absicherung			Speed
		AP $^{50:.05:95}_{COCO}$	AP $^{50}_{PASCAL\ VOC}$	LAMR ^{50}	AP $^{50:.05:95}_{COCO}$	AP $^{50}_{PASCAL\ VOC}$	LAMR ^{50}	Frames/Sec
FasterRCNN [33]	2D-OD	0.51	0.81	0.61	0.57	0.90	0.51	14.65
FCOS [38]	2D-OD	0.56	0.83	0.38	0.61	0.90	0.32	16.16
RetinaNet [20]	2D-OD	0.50	0.80	0.42	0.52	0.86	0.40	15.35
SSD300 [22]	2D-OD	0.18	0.44	0.83	0.21	0.54	0.86	97.08
MaskRCNN [14]	Ins. Seg.	0.52	0.80	0.61	0.54	0.89	0.51	9.99
KeypointRCNN [14]	KP 2D	–	–	–	0.58	0.90	0.50	4.03

Table 1: The training results of the pedestrian detection networks on different tasks and datasets. The tasks include: 2D object detection (2D-OD), semantic instance segmentation (Ins. Seg.) and keypoint detection within 2D object detection (KP 2D). The trained networks include FasterRCNN [33], FCOS [38], RetinaNet [20], SSD [22] with the input image size of 300 × 300, MaskRCNN and KeypointRCNN [14]. The evaluation metrics include the COCO [19] evaluation metric with an IoU threshold range from 0.5 to 0.95 with the step size of 0.05 (higher is better), the PASCAL VOC [10] metric with an IoU threshold of 0.5 (higher is better) and the Log-Average Miss Rate (LAMR) [9] metric with an IoU threshold of 0.5 (lower is better). The trainings are done on two datasets, namely the CityPersons [43] and the KI-Absicherung [1] dataset.

4 Experimental Setup

In this section, the experiment setup for our experiments are introduced. We have implemented and trained six of the state-of-the-art detectors for pedestrian detection within automated driving scenarios with three different sub-tasks, including:

- **2D Object Detection:** which we trained the well known FasterRCNN by Ren *et al*. [33], the FCOS by Tian *et al*. [38], the RetinaNet by Lin *et al*. [20] and the SSD by Liu *et al*. [22].
- **Semantic Instance Segmentation:** which we trained the MaskRCNN network by He *et al*. [14].
- **Key-point Detection:** which we trained the KeypointRCNN by He *et al*. [14].

The instance segmentation and keypoint detection networks are from the well known RCNN family which have similar architectures compared to FasterRCNN. We chose to employ those networks since they offer the possibility to extract (dis-)similarities when only changing the detection heads, therefore enabling us to comparatively analyze PLFs of highly similar network architectures across different tasks. All of the aforementioned networks were trained on the synthetic dataset produced in the KI-Absicherung project [1] with more than 200,000 samples on a ResNet [13] backbone with 50 layers, which was pre-trained on the ImageNet dataset [8]. Furthermore, all of the aforementioned networks except the KeypointRCNN were also trained on the publicly available CityPersons real dataset [43] which is extracted from the Cityscapes [7] semantic segmentation dataset with 3500 samples. For training the aforementioned DNNs on the CityPersons dataset, we used the pre-trained weights from the KI-Abischerung dataset. The training details as well as the parameters are discussed in the Supplementary Section Supp.1. Furthermore, we have limited the distance of the pedestrians to the camera to be below fifty meters in our evaluations. The reason behind this decision was that the KI-Absicherung dataset is generated mainly in the urban traffic area and there are considerable amount of pedestrians in far distances that challenged the evaluation of the DNN. This decision makes it easier to evaluate factors other than distance, which is clearly a PLF from previous studies. Consequentially, for the purpose of

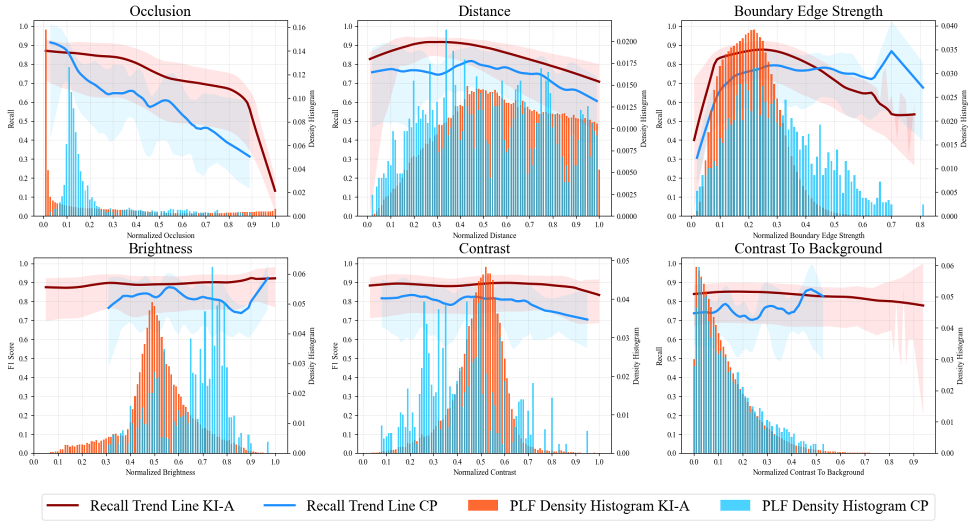

Figure 2: The correlation results of the three of the factors to be considered as PLF (top row) and three factors to not be considered as PLF (bottom row) averaged over all the six trained models on KI-Absicherung dataset and the five models trained on the CityPersons dataset. The x-axis represents the normalized values of the factors, and the y-axis the recall. The lines represent the local regression correlation of each factor and the according local performance. The histogram is calculated upon the frequency of each factor value in the respective dataset.

an equivalent evaluation, we have also limited the pedestrians presented in the CityPerson dataset to be within the fifty meters distance from the camera.

5 Results and Discussion

In this section, we discuss our observations after analyzing the results of the aforementioned experiments. The results of the training are presented in Table 1. One can observe from this table that different DNNs achieved similar results on both of the datasets except for the SSD300, which is a much smaller and faster DNNs compared to the others, and achieved the least performance among the all. The training parameters and details per each DNNs are presented in the supplementary section. Examples of images with different factors are shown in the Figure 1. These include distance, wetness, fog, brightness vignette and contrast. As one can observe from this images, there is a probability that one sample image can have different levels of one individual factors in different regions, i.e. high contrast in the sun and low contrast in the shadow on the same image. There is also the chance that an image might have different factors all together, *e.g.* crowdedness and occlusion. Such examples make the analysis of a single PLF on a single image hard, as there is no strategy to distinguish the contribution of each PLF to the possible low performance of the DNNs in one single sample. Based on that, we argue that, in order to understand the effect of a single PLF on a DNNs performance, one needs to evaluate it globally over the whole dataset rather than on individual data samples. This includes visualizing different correlation graphs and calculating the correlation coefficient of different factors with the networks performance. While studying the performance of the DNNs w.r.t. the factors that are calculated per each object, calculating precision and other metrics that require false positives would be impossible. Therefore, we suggest to consider only recall as the standard metric for studying the performance correlation with such factors. For the factors that are extracted per image, we report on both the recall and the f1 score.

Correlation Graphs: the results on the correlation graphs can be found in Figure 2. As one can observe from this graph, the results are shown for three of the factors that can be considered as PLF (top row) and three which do not follow our definition and criteria of a PLF (bottom row). These include object occlusion, object distance, object boundary edge strength, image brightness, image contrast and object contrast to background. It can be observed from this figure that the histogram of the individual factor levels are different between our real-world and synthetic dataset. In other words, the real-world dataset has a more diverse histogram, while the synthetic one is either normal or skewed to left or right, having a unimodal shape. On the other hand, one can observe that the object occlusion has a higher correlation with performance. Based on our definition of a PLF in 3, we argue that in order for a factor to be a PLF, there should be a strong correlation between the performance and the factor on one hand, and no observable correlation between the factor frequency histogram and the performance on the other hand. Particularly, the correlation between the factor and the performance should not stem from the under/over representation of data leading to under-/overfitting of the network to specific data properties. Based upon this, one can observe that the aforementioned conditions are observable for the top three factors illustrated in Figure 2, while not observable for the bottom three. Furthermore, after qualitative inspections on the individual DNNs which can be found in Supplementary Section Supp.2, we did not observe significant differences in the correlation trend among different tasks. More specifically, as we have trained the semantic instance segmentation and keypoint detection DNNs from the RCNN architecture, we compared these two with the FasterRCNN DNN for 2D object detection and observed no significant difference in performance trend among different factors.

The correlation coefficient results of all the studied factors are illustrated in the Figure 3. As the number of total factors extracted for the CityPersons dataset are lower than the KI-Absicherung dataset, a direct comparison of all the factors among the two datasets was not possible. Firstly, one can observe that there are various factors that have a very similar correlation coefficient for both of the datasets, *e.g.* occlusion, entropy, contrast, *etc.*, while there are other factors that have a bigger difference, such as contrast to background, foreground brightness and edge strength. However, as the correlation coefficient is not the only metric to assess for the existence of a PLF, we also utilized a qualitative inspection of the graphs such as the ones illustrated in the Figure 2 to evaluate the factors for PLFs. Based on that, one can observe that some factors such as boundary edge strength can be considered as PLF despite their low correlation coefficient. This is only observable by such a qualitative inspection, where one can realize that there exists a strong correlation with the respective factor while having a general low correlation coefficient. Consequently, we selected the occlusion, distance, boundary edge strength (ours), background edge strength (ours), height, crowdedness (ours), edge strength, fog intensity, and wetness types as PLF, and disregarded all the others factors as non-PLFs. Based on this analysis, occlusion had the most significant effect among the selected metrics as PLF. However, we do not suggest to immediately reject the second group of factors before enhancing the training datasets with more samples of the respective factor to equalize the histogram distribution and re-assessing their effect on DNNs performance. This could possibly lead to alleviation of the effect of some of the factors that

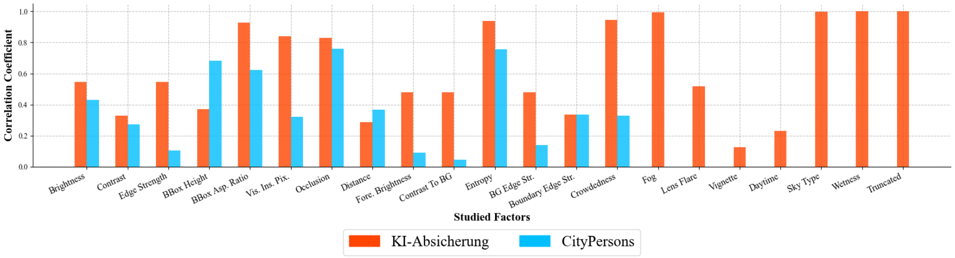

Figure 3: The results of the correlation coefficients calculated between the studied factors and the models recall values, averaged over all the trained models. X-axis include all the studied factors, while y-axis the according correlation coefficient values. The factors from left to right include brightness, contrast, edge strength, bounding box height, bounding box aspect ratio, visible instance pixels, occlusion ratio, distance, foreground brightness, contrast to background, object entropy, background edge strength, boundary edge strength, object crowdedness, fog intensity, lens flare intensity, daytime, sky type, wetness, and truncated. Orange bars represent the results from the KI-Absicherung dataset, and the blue bars represent the results from the CityPersons dataset.

stem from dataset bias.

This paper is one of the first studies to investigate the impact of multiple factors on the performance of DNNs systematically. Out of these experiments, we studied multiple DNNs with three different tasks. We encourage further research with other DNN architectures to investigate the consistency of PLFs across different architectures and different tasks and to introduce thresholds on the losses in performance. Along with enhancing the training datasets using the aforementioned methods to remove the dataset bias w.r.t. different factors, other future works would include optimization techniques and loss functions to alleviate the effect of such factors. Finally, the authors suggest to also investigate multiple performance metrics as well as other tasks to extract and study the effect of such factors on the DNNs performance.

6 Conclusions

In this paper, we discussed the concept of the DNNs performance limiting factors (PLF) and proposed an approach to define a factor as a PLF. Based on that, we have analyzed seventeen factors from the literature alongside with four novel factors, out of which nine can be considered as PLF. We have analyzed six state-of-the-art DNNs that cover three tasks including 2D object detection, semantic instance segmentation and keypoint detection trained and tested on a large scale synthetic dataset and a smaller real-world dataset. We have shown in our experiments that regardless of the data distribution, a PLF could always hinder the DNNs to predict accurately. We have shown that such PLFs affect the DNNs even across different tasks and with different architectures similarly.

Acknowledgement

The research leading to the results presented above are funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for automated driving". We thank Prof. Visvanathan Ramesh for the technical supervision of this work. Prof. Visvanathan Ramesh acknowledges the support of the Artificial Intelligence Systems Engineering Laboratory (AISEL) project under funding number 01IS19062, funded by the German Federal Ministry of Education and Research (BMBF).

References

[1] KI Absicherung – Safe AI for automated driving. https://www.ki-absicherung-projekt.de/. Accessed: 2022-10-06.

[2] Yasin Bayzidi, Alen Smajic, Fabian Hüger, Ruby Moritz, Serin Varghese, Peter Schlicht, and Alois Knoll. Traffic sign classifiers under physical world realistic sticker occlusions: A cross analysis study. In *33rd IEEE Intelligent Vehicles Symposium (IV)*, Jun 2022.

[3] Alexander von Bernuth, Georg Volk, and Oliver Bringmann. Simulating photo-realistic snow and fog on existing images for enhanced cnn training and evaluation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 41–46, 2019. doi: 10.1109/ITSC.2019.8917367.

[4] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11679–11689, 2020. doi: 10.1109/CVPR42600.2020.01170.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Keenan Burnett, Andreas Schimpe, Sepehr Samavi, Mona Gridseth, Chengzhi Winston Liu, Qiyang Li, Zachary Kroeze, and Angela P. Schoellig. Building a winning self-driving car in six months. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9583–9589, 2019. doi: 10.1109/ICRA.2019.8794029.

[7] 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 CVPR*, pages 3213–3223, Las Vegas, NV, USA, June 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[9] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012. doi: 10.1109/TPAMI.2011.155.

[10] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88 (2):303–338, jun 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4. URL https://doi.org/10.1007/s11263-009-0275-4.

[11] Lydia Gauerhof, Richard Hawkins, Chiara Picardi, Colin Paterson, Yuki Hagiwara, and Ibrahim Habli. Assuring the safety of machine learning for pedestrian detection at crossings. In António Casimiro, Frank Ortmeier, Friedemann Bitsch, and Pedro Ferreira, editors, *Computer Safety, Reliability, and Security*, pages 197–212, Cham, 2020. Springer International Publishing.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[15] Shih-Chia Huang, Trung-Hieu Le, and Da-Wei Jaw. Dsnet: Joint semantic learning for object detection in inclement weather conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2623–2633, 2021. doi: 10.1109/TPAMI.2020.2977911.

[16] Nikhil Kapoor, Chun Yuan, Jonas Löhdefink, Roland Zimmerman, Serin Varghese, Fabian Hüger, Nico Schmidt, Peter Schlicht, and Tim Fingscheidt. A self-supervised feature map augmentation (fma) loss and combined augmentations finetuning to efficiently improve the robustness of cnns. In *Computer Science in Cars Symposium*, CSCS ’20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376211. doi: 10.1145/3385958.3430477. URL https://doi.org/10.1145/3385958.3430477.

[17] Nikhil Kapoor, Andreas Bär, Serin Varghese, Jan David Schneider, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. From a fourier-domain perspective on adversarial examples to a wiener filter defense for semantic segmentation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9534145.

[18] Mourad A. Kenk and M. Hassaballah. Dawn: Vehicle detection in adverse weather nature dataset. *ArXiv*, abs/2008.05402, 2020.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

[21] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6452–6461, 2019. doi: 10.1109/CVPR.2019.00662.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[23] Maria Lyssenko, Christoph Gladisch, Christian Heinzemann, Matthias Woehrle, and Rudolph 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*, pages 38–45, June 2021.

[24] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming, 2020. URL https://openreview.net/forum?id=ryljMpNtwr.

[25] Valentina Musat, Ivan Fursa, Paul Newman, Fabio Cuzzolin, and Andrew Bradley. Multi-weather city: Adverse weather stacking for autonomous driving. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2906–2915, 2021. doi: 10.1109/ICCVW54120.2021.00325.

[26] Andreas Nussberger, Helmut Grabner, and Luc Van Gool. Robust aerial object tracking in images with lens flare. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6380–6387, 2015. doi: 10.1109/ICRA.2015.7140095.

[27] Isaac Ogunrinde and Shonda Bernadin. A review of the impacts of defogging on deep learning-based object detectors in self-driving cars. In *SoutheastCon 2021*, pages 01–08, 2021. doi: 10.1109/SoutheastCon45413.2021.9401941.

[28] 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*, 12:2825–2830, 2011.

[29] giovanni pepe, leonardo gabrielli, livio ambrosini, stefano squartini, and luca cattani. detecting road surface wetness using microphones and convolutional neural networks. *journal of the audio engineering society*, march 2019.

[30] Xiaotian Qiao, Gerhard P. Hancke, and Rynson W.H. Lau. Light source guided single-image flare removal from unpaired data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4177–4185, October 2021.

[31] Yangyang Qu, Yongsheng Ou, and Rong Xiong. Low illumination enhancement for object detection in self-driving. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1738–1743, 2019. doi: 10.1109/ROBIO49542.2019.8961471.

[32] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. Failing to learn: Autonomously identifying perception failures for self-driving cars. *IEEE Robotics and Automation Letters*, 3(4):3860–3867, 2018. doi: 10.1109/LRA.2018.2857402.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[34] Kmeid Saad and Stefan-Alexander 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)*, pages 1–5, 2019. doi: 10.1109/ICCVE45908.2019.8965233.

[35] Miguel Ángel Sotelo, Francisco Javier Rodríguez, L. Magdalena, Luis Miguel Bergasa, and Luciano Boquete. A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, 16:95–116, 2004.

[36] Izzeddin Teeti, Valentina Musat, Salman Khan, Alexander Rast, Fabio Cuzzolin, and Andrew Bradley. Vision in adverse weather: Augmentation using cyclegans with various object detectors for robust perception in autonomous racing. *Workshop on Safe Learning for Autonomous Driving at ICML 2022*, abs/2201.03246, 2022. URL https://learn-to-race.org/workshop-sl4ad-icml2022/assets/papers/paper_17.pdf.

[37] Binyu Tian, Felix Juefei-Xu, Qing Guo, Xiaofei Xie, Xiaohong Li, and Yang Liu. Ava: Adversarial vignetting attack against visual recognition. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1046–1053. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019. doi: 10.1109/ICCV.2019.00972.

[39] Serin Varghese, Yasin Bayzidi, Andreas Bar, Nikhil Kapoor, Sounak Lahiri, Jan David Schneider, Nico M. Schmidt, Peter Schlicht, Fabian Huger, and Tim Fingscheidt. Unsupervised temporal consistency metric for video segmentation in highly-automated driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[40] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7774–7783, 2018. doi: 10.1109/CVPR.2018.00811.

[41] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan T. Barron. How to train neural networks for flare removal. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2219–2227, 2021. doi: 10.1109/ICCV48922.2021.00224.

[42] Keisuke Yoneda, Naoki Suganuma, Ryo Yanase, and Mohammad Aldibaja. Automated driving recognition technologies for adverse weather conditions. *IATSS Research*, 43 (4):253–262, 2019. ISSN 0386-1112. doi: https://doi.org/10.1016/j.iatssr.2019.11.005. URL `https://www.sciencedirect.com/science/article/pii/S0386111219301463`.

[43] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4457–4465, 2017. doi: 10.1109/CVPR.2017.474.

Supplementary Information

Supp.1 Training Parameters

The tables of the training parameters for all the DNNs trained on the KI-Absicherung [1] dataset are presented in Table S.1.1 and the CityPersons [43] dataset in Table S.1.2.

Trained Model	Best Epoch	Batch Size	Optimizer	Learning Rate	Weight Decay	Momentum	LR Scheduler Step Size	LR Scheduler Gamma
FasterRCNN	10	4	SGD	0.01	0.0001	0.9	5	0.1
FCOS	8	4	SGD	0.01	0.0001	0.9	5	0.1
RetinaNet	14	4	SGD	0.001	0.0001	0.9	5	0.1
SSD300	11	4	SGD	0.001	0.0001	0.9	5	0.1
KeypointRCNN	8	4	SGD	0.01	0.0001	0.9	5	0.1
MaskRCNN	12	4	SGD	0.01	0.0001	0.9	5	0.1

Table S.1.1: The training parameters of all the trained DNNs on the KI-Absicherung [1] dataset. From left to right the columns include: Trained model; best epoch, on which the training stopped afterwards due to no significant improvement; optimizer, which is stochastic gradient descent for all the DNNs; learning rate; weight decay used for regularization; momentum; learning rate scheduler step size used to decrease the learning rate size after so many steps; learning rate scheduler gamma used to decrease the learning rate by that factor. All the DNNs are initially set to train up to 50 epochs but stopped earlier due to the early stopping condition being fulfilled.

Trained Model	Best Epoch	Batch Size	Optimizer	Learning Rate	Weight Decay	Momentum	LR Scheduler Step Size	LR Scheduler Gamma
FasterRCNN	17	4	SGD	0.01	0.0001	0.9	10	0.1
FCOS	9	4	SGD	0.001	0.0001	0.9	10	0.1
RetinaNet	20	4	SGD	0.001	0.0001	0.9	10	0.1
SSD300	49	4	SGD	0.001	0.0001	0.9	10	0.1
MaskRCNN	20	4	SGD	0.01	0.0001	0.9	10	0.1

Table S.1.2: The training parameters of all the trained DNNs on the CityPersons [43] dataset. From left to right the columns include: Trained model; best epoch, on which the training stopped afterwards due to no significant improvement; optimizer, which is stochastic gradient descent for all the DNNs; learning rate; weight decay used for regularization; momentum; learning rate scheduler step size used to decrease the learning rate size after so many steps; learning rate scheduler gamma used to decrease the learning rate by that factor. All the DNNs are initially set to train up to 50 epochs but stopped earlier due to the early stopping condition being fulfilled.

Supp.2 Correlation Results Per Factors

The figures of all the studied factors averaged over all the DNNs are presented in this supplementary section. The results are grouped into the PLFs and non-PLFs. Based on that, the factors object occlusion in Figure S.2.1, object distance in Figure S.2.2, image edge strength in Figure S.2.3, object boundary edge strength in Figure S.2.4, object background edge strength in Figure S.2.5, object bounding box height in Figure S.2.6, object crowdedness in Figure S.2.7, image fog intensity in Figure S.2.8, and image wetness type in Figure S.2.9 are presented as PLFs. Furthermore, the factors image contrast in Figure S.2.10, object contrast to background in Figure S.2.11, image brightness in Figure S.2.12, object foreground brightness in Figure S.2.13, object entropy in Figure S.2.14, object bounding box aspect ratio in Figure S.2.15, object visible instance pixels in Figure S.2.16, image lens flare intensity in Figure S.2.17, image vignette intensity in Figure S.2.18, image daytime type in Figure S.2.19, image sky type in Figure S.2.20 and object truncated in Figure S.2.21 are presented as non-PLFs factors.

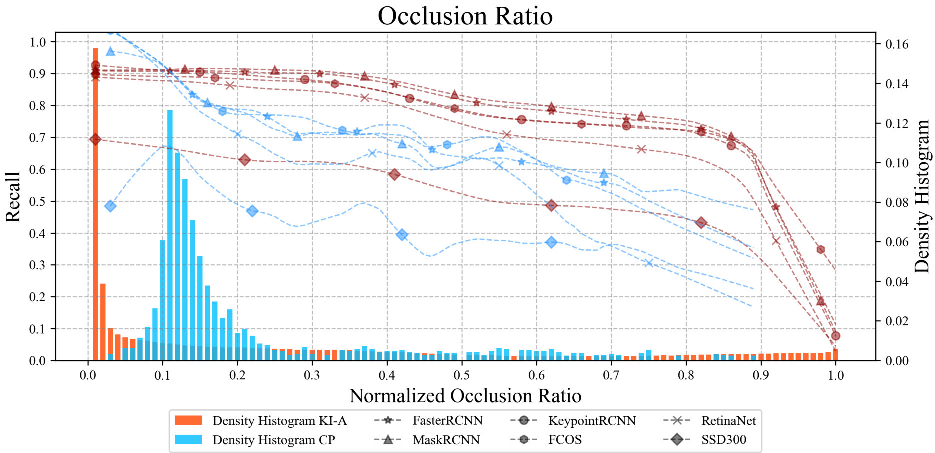

Figure S.2.1: The correlation results of the object occlusion ratio averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

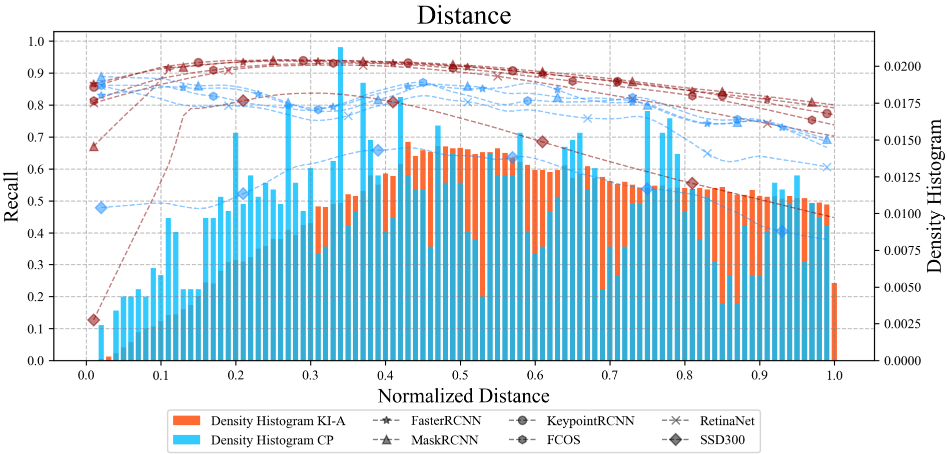

Figure S.2.2: The correlation results of the object distance averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

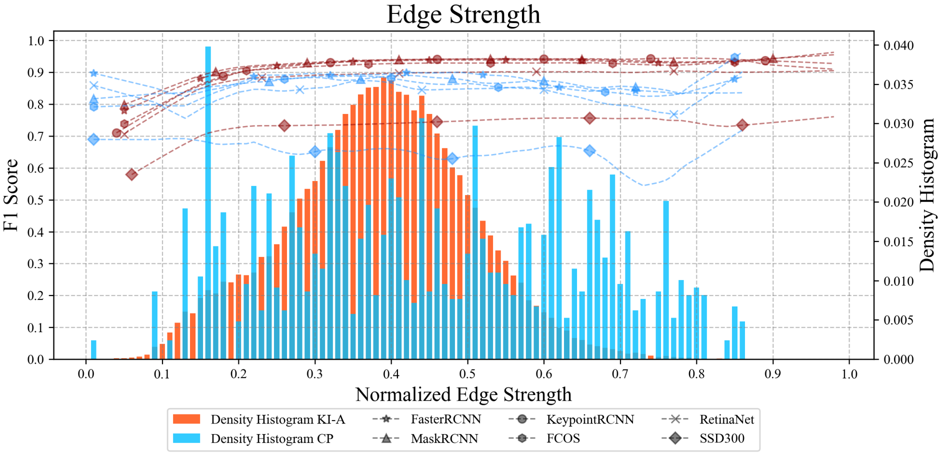

Figure S.2.3: The correlation results of the image edge strength averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

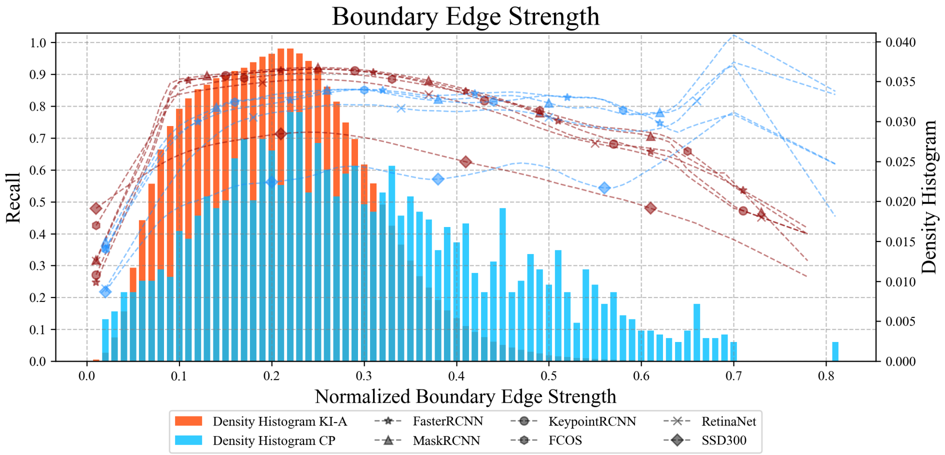

Figure S.2.4: The correlation results of the object boundary edge strength averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

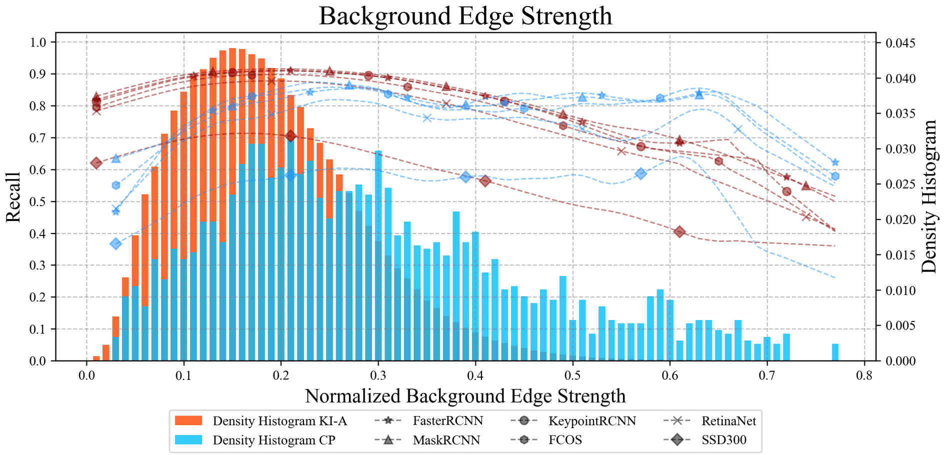

Figure S.2.5: The correlation results of the object background edge strength averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

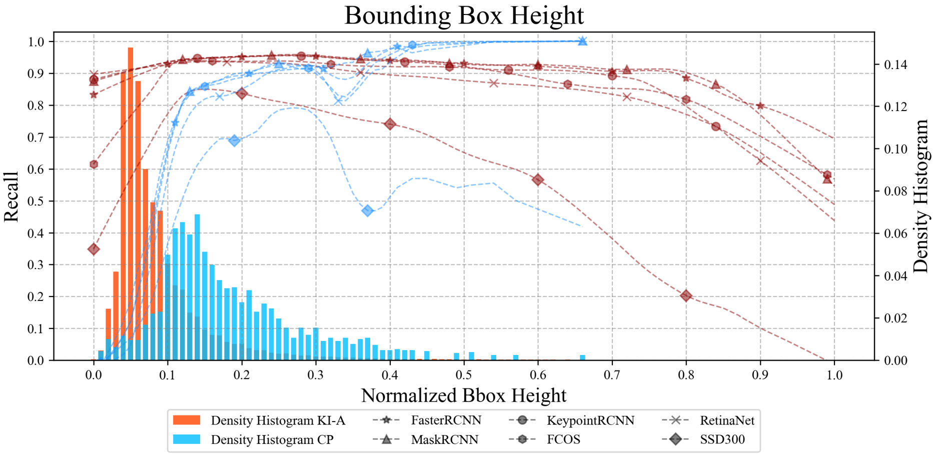

Figure S.2.6: The correlation results of the object bounding box height averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

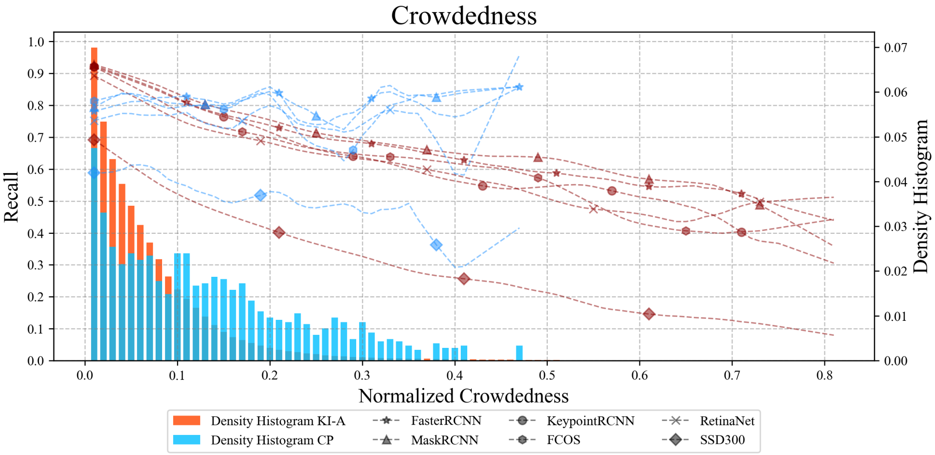

Figure S.2.7: The correlation results of the object crowdedness averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

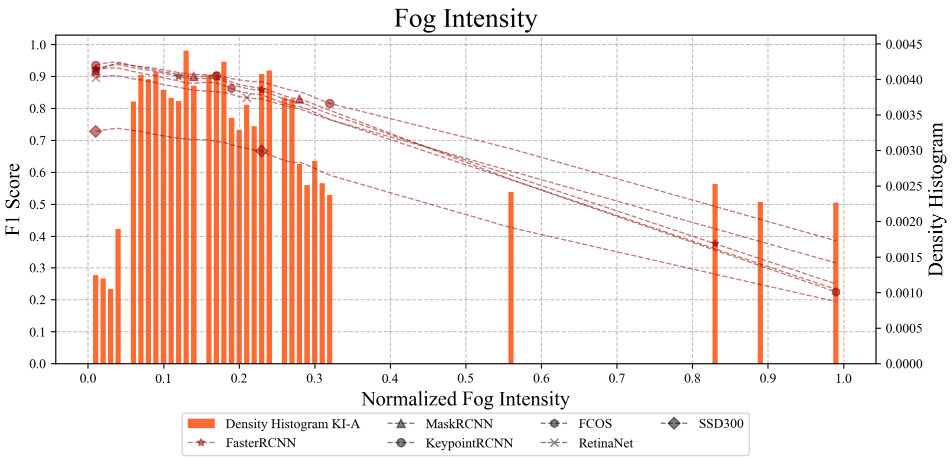

Figure S.2.8: The correlation results of the image fog intensity averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

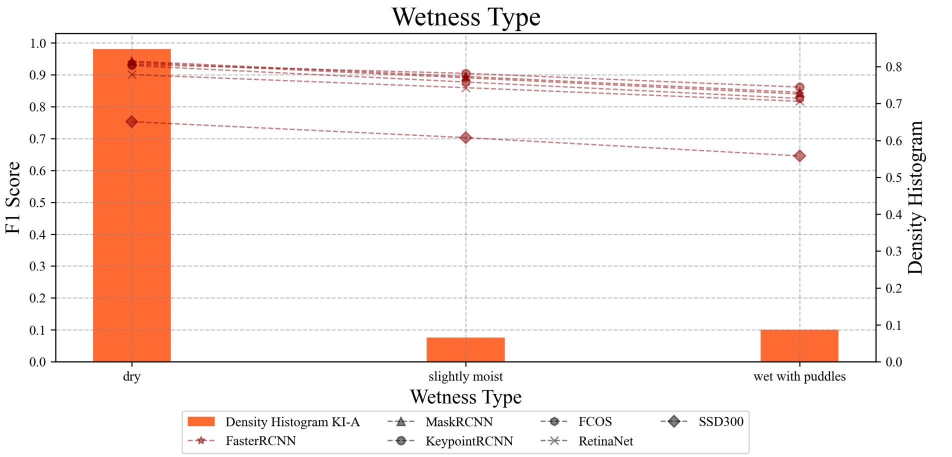

Figure S.2.9: The correlation results of the image wetness type averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

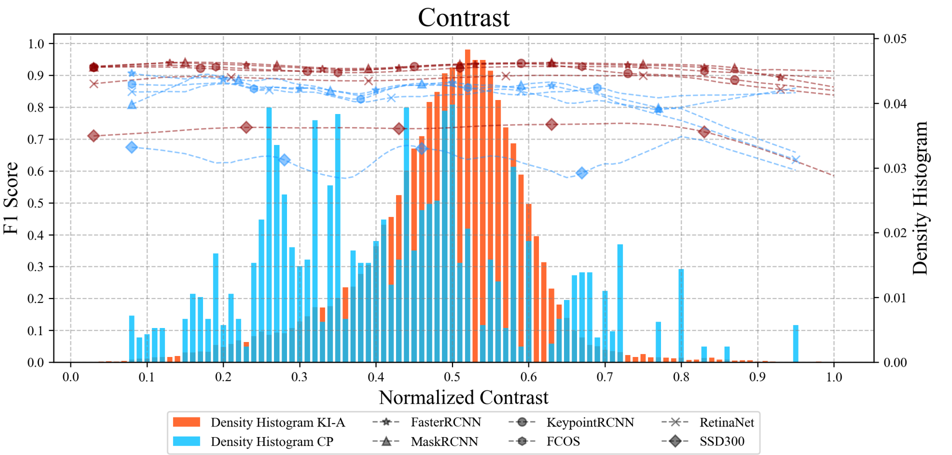

Figure S.2.10: The correlation results of the image contrast averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

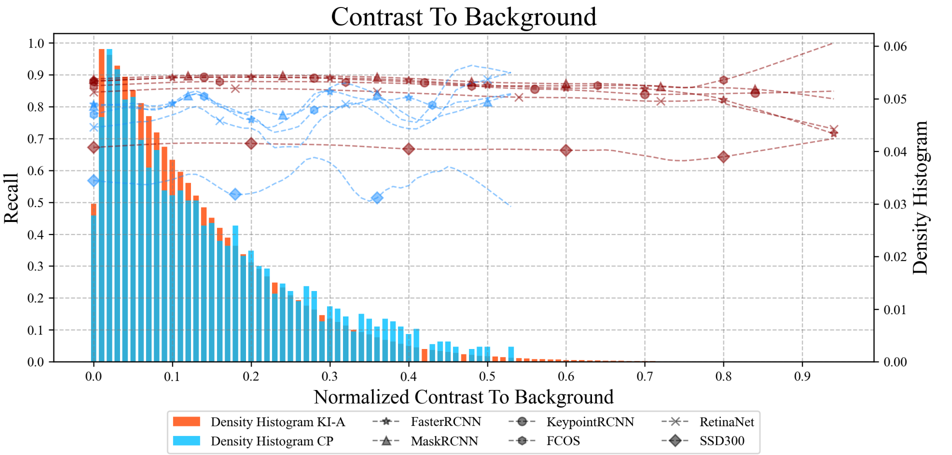

Figure S.2.11: The correlation results of the object contrast to background averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

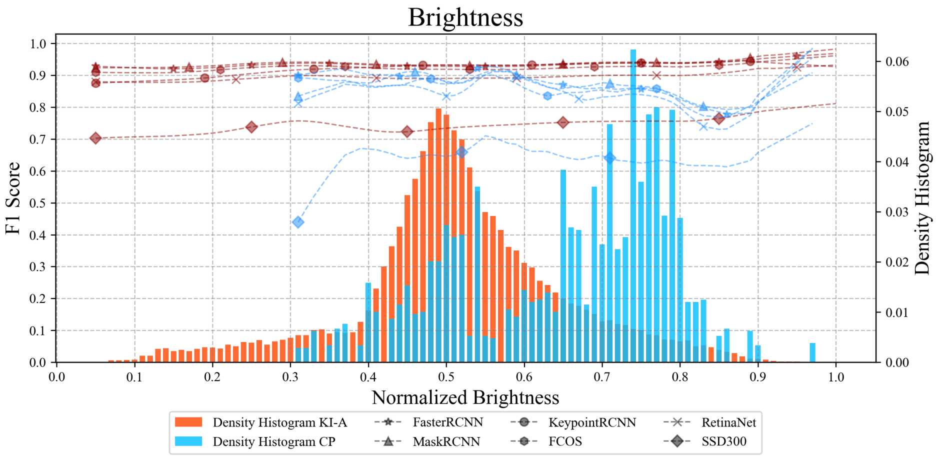

Figure S.2.12: The correlation results of the image brightness averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

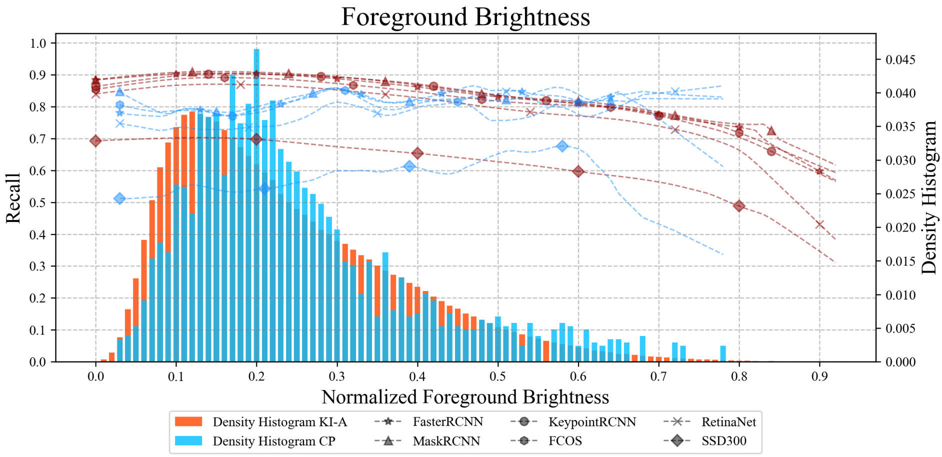

Figure S.2.13: The correlation results of the object foreground brightness averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

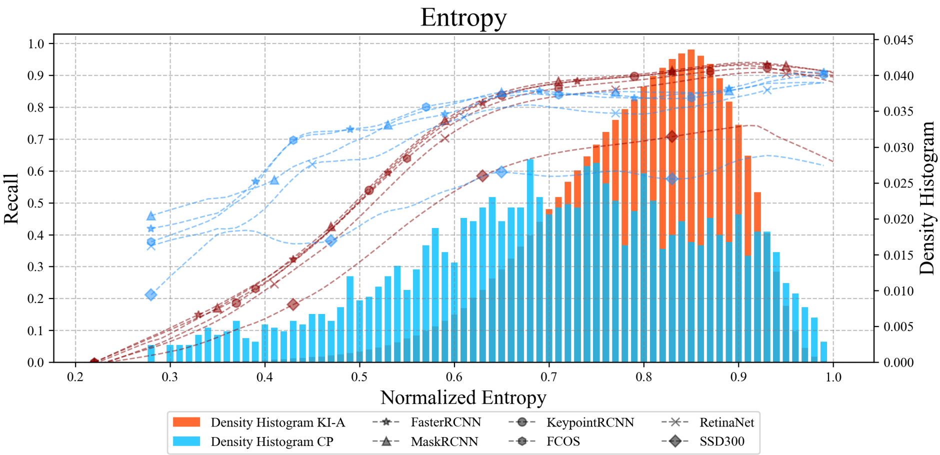

Figure S.2.14: The correlation results of the object entropy averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

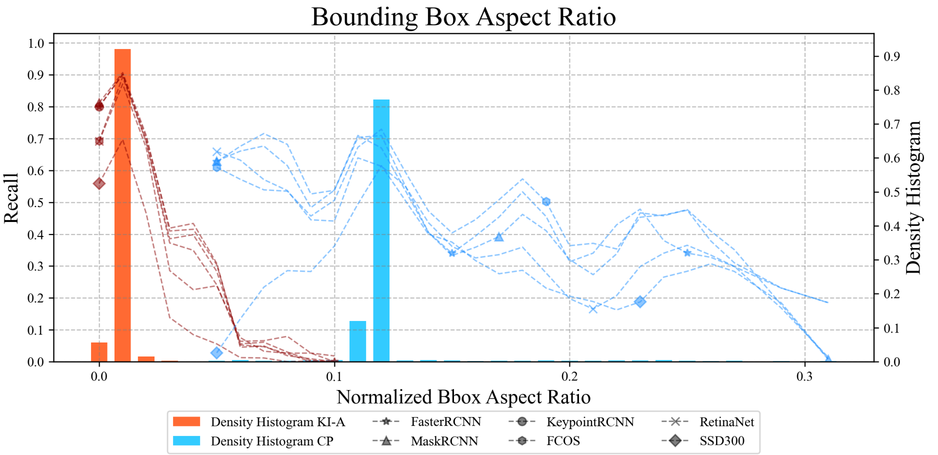

Figure S.2.15: The correlation results of the object bounding box aspect ratio averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

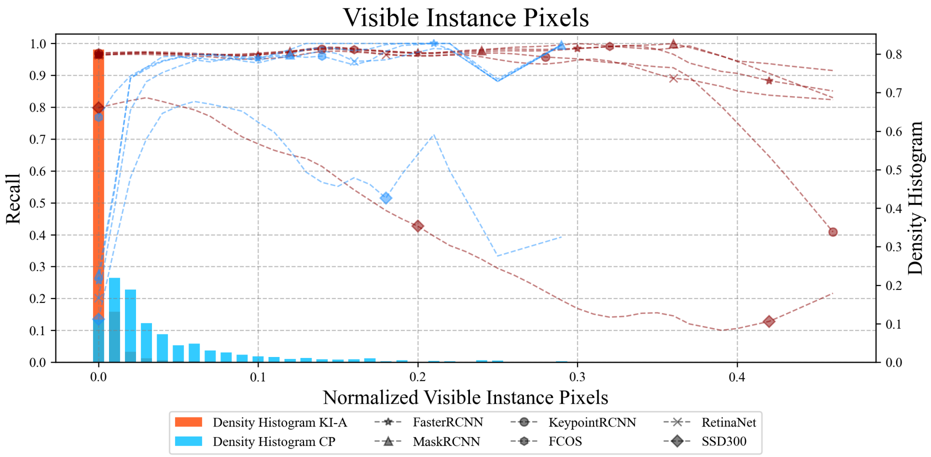

Figure S.2.16: The correlation results of the object visible instance pixels averaged over all the six trained models on KI-Absicherung dataset and the five trained models on the CityPersons dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The lines represent the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

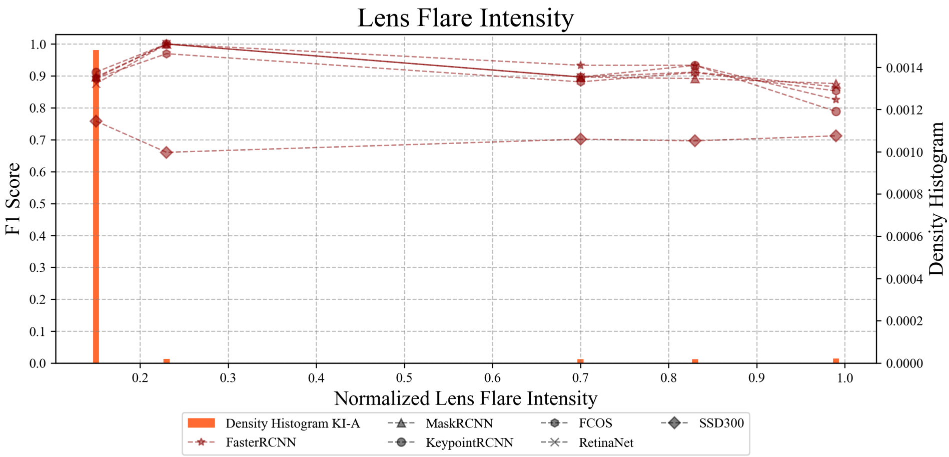

Figure S.2.17: The correlation results of the image lens flare intensity averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

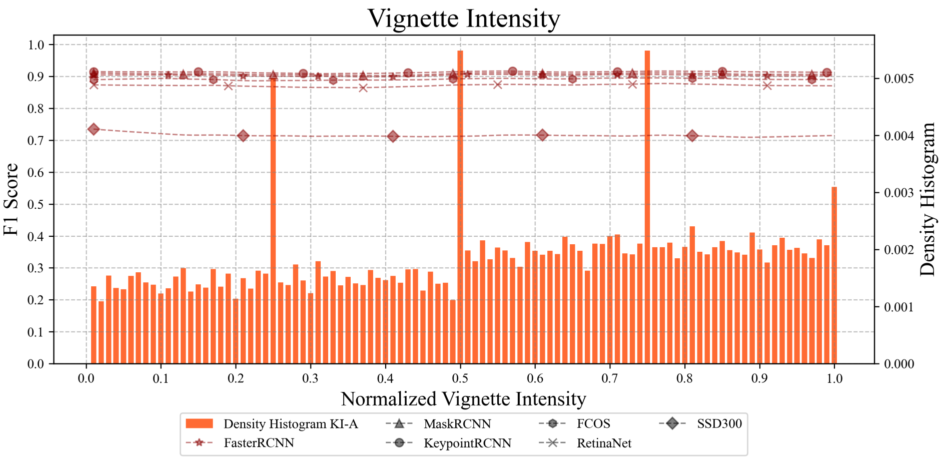

Figure S.2.18: The correlation results of the image vignette intensity averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

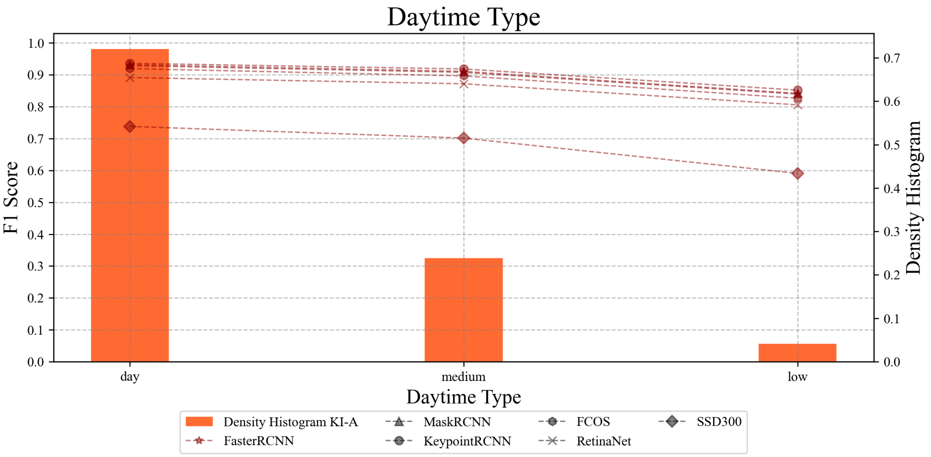

Figure S.2.19: The correlation results of the image daytime type averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

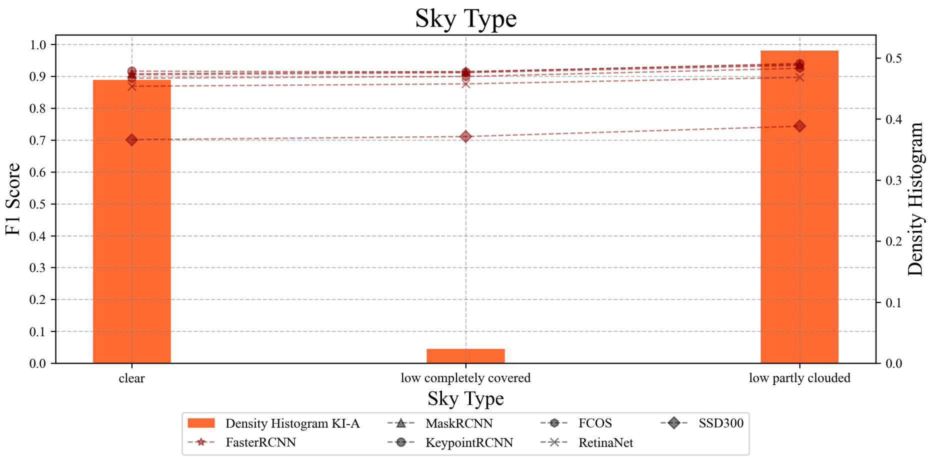

Figure S.2.20: The correlation results of the image sky type averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.

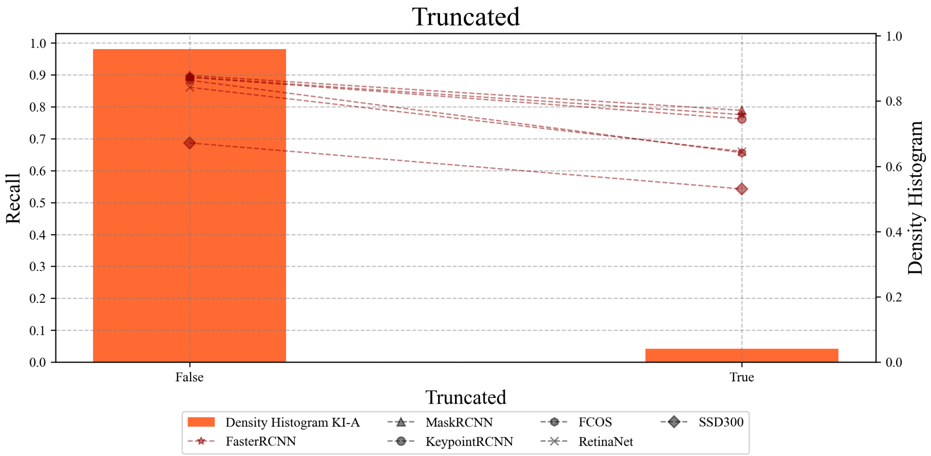

Figure S.2.21: The correlation results of the object truncated averaged over all the six trained models on KI-Absicherung dataset. The x-axis represents the normalized values of this factor, and the y-axis the recall. The line represents the local regression correlation of this factor and the according local performance and the bars represent the histogram of this factor frequency in the according level.