Safety Monitoring for Pedestrian Detection in Adverse Conditions

Swapnil Mallick, Shuvam Ghosal(✉), Anand Balakrishnan, and Jyotirmoy Deshmukh

University of Southern California, Los Angeles, CA 90007, USA
{smallick,sghosal,anandbal,jdeshmuk}@usc.edu

Abstract. Pedestrian detection is an important part of the perception system of autonomous vehicles. Foggy and low-light conditions are quite challenging for pedestrian detection, and several models have been proposed to increase the robustness of detections under such challenging conditions. Checking if such a model performs well is largely evaluated by manually inspecting the results of object detection. We propose a monitoring technique that uses Timed Quality Temporal Logic (TQTL) to do differential testing: we first check when an object detector (such as vanilla YOLO) fails to accurately detect pedestrians using a suitable TQTL formula on a sequence of images. We then apply a model specialized to adverse weather conditions to perform object detection on the same image sequence. We use Image-Adaptive YOLO (IA-YOLO) for this purpose. We then check if the new model satisfies the previously failing specifications. Our method shows the feasibility of using such a differential testing approach to measure the improvement in quality of detections when specialized models are used for object detection.

Keywords: Pedestrian Detection · Autonomous Driving · Temporal Logic

1 Introduction

Convolutional Neural Network (CNN) based models [5,10,14,25,28] have become widespread in the field of pedestrian detection and have been able to achieve impressive results when tested on benchmark driving datasets. Some of these models have also been deployed in autonomous vehicles [29].

According to the California DMV statistics, 627 autonomous vehicle collision have been reported as of July 25, 2023 despite numerous object detection models faring well when tested on high-quality images captured in clear weather conditions [1]. An important question is how these models fare under challenging lighting conditions such as when it is foggy or dark. Beyond the obvious issues presented by images being out-of-distribution with respect to the lighting conditions, this problem becomes harder when pedestrians wear black or dark colored clothes at night.

S. Mallick and S. Ghosal—These authors contributed equally to this work.

P. Katsaros and L. Nenzi (Eds.): RV 2023, LNCS 14245, pp. 1–11, 2023.
https://doi.org/10.1007/978-3-031-44267-4_22

In this case study paper, we have two main objectives. First, we wish to demonstrate a logic-based metric that can show that pedestrian detection in low-light conditions suffers in quality *without the use of ground-truth annotations*. Towards this end, we use the YOLO object detection model [24] trained on a standard driving dataset, and apply it to videos containing poor lighting conditions. Our technical idea is to use the quality metric of Timed Quality Temporal Logic TQTL [4,7]. We show that we can express object consistency properties in TQTL, and use the PerceMon tool [3] to monitor violations of the given TQTL specification by the sequence of detections output by YOLO.

We then use a modified object detection framework called IA-YOLO [17]. IA-YOLO has a component that predicts parameters to be used for a differentiable image processing (DIP) module. The DIP module can be thought of as a way to implement a number of image filters to improve performance of object detection. We then monitor the detections by IA-YOLO against the same TQTL specification.

The main conclusions of this study are as follows: (1) The quality of a TQTL specification on detections using vanilla is poor when pre-trained YOLO is used on videos with low-light conditions. (2) The quality of TQTL specifications is significantly improved when IA-YOLO is used for detections.

Related Work. Significant research has been conducted for general pedestrian detection. However, only a few efforts have been made to detect pedestrians successfully in adverse lighting and weather conditions. Works like IA-YOLO [17] and DENet [22] use filters implementing ideas similar to those used in image processing to preprocess images for object detectors like YOLO, and thereby improving process. Other works, like the CycleGAN framework presented in [27] augment existing datasets with generative images that simulate adverse weather conditions to improve the robustness of an object detector. In a similar light, the authors of [18] present datasets synthetically generated images that simulate adverse weather conditions.

Evaluation of multi-object detection models have predominantly used the *mean average precision* (mAP) metric, popularized by the PASCAL-VOC dataset [8]. This metric evaluates the accuracy of object detectors by estimating the area under the curve of the precision-recall relationship with respect to a given dataset. Mean average precision(mAP) is the average of the Average precision(AP) values for each output object class. While other metrics have been proposed in literature, these aren't as popular as the mAP evaluation metric [20,21].

The above approaches rely on either augmenting existing datasets to improve robustness of object detectors, or comparing against ground truth to evaluate the accuracy of a model. To this light, recent literature has proposed the use of temporal logics to evaluate perception systems without access to groundtruth data [4,6,12]. These use TQTL (and its extension Spatio-temporal Quality Logic) to monitor for incorrect behavior of object detector models. Similarly,

the works in [3] and [2] propose the use of online monitoring techniques to perform fault detection at runtime.

2 Preliminaries

Object detection is a computer vision task to check the presence of objects of a certain class in an image and also obtain a bounding box (in image coordinates) indicating the location of the object within the image. Object detection algorithms can be broadly categorized into two groups based on how they operate. Some algorithms propose regions of interest (RoIs) in the image space [9] and then classify the regions by training neural networks. These algorithms are called the region proposal-based methods. The other class of algorithms comprises single-stage regression based methods [15,16], such as the YOLO series [23] of algorithms. In algorithms like YOLO, the result of applying object detection to an image is a list of bounding boxes in image coordinates, and for each bounding box in this list, we obtain a class label (for the purported object in the box), and a number in $[0, 1]$ that indicates the confidence of the detector in the class label. In autonomous driving applications, class labels may include *bicycles*, *cars*, *pedestrians*, *traffic lights*, etc.

Usually, images captured in adverse weather or during low-light conditions do not have the same distribution of low-level features, which leads to poor detection of some kinds of object classes. Some weather effects like rain or fog can obscure key features or have the effect of adding noise to the image. In traditional image processing literature, custom image filters can be designed to denoise images [30], remove the effects of certain weather phenomena [19,26], or enhance features required for detection. However, designing custom image filters is a manual and tedious process. An approach to overcome this problem is to use an adaptive detection model such as IA-YOLO. Such a model can filter out weather-specific information and highlight latent information to make detection easier. We discuss IA-YOLO next.

2.1 IA-YOLO

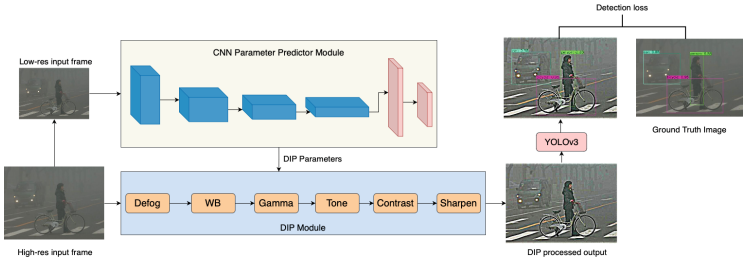


Fig. 1. The training pipeline of the IA-YOLO framework.

The IA-YOLO pipeline consists of (1) a parameter predictor based on a convolutional neural network (CNN), (2) a differentiable image processing module (DIP), and a (3) detection network. Before applying the IA-YOLO pipeline, we resize a given image to have 256×256 resolution, and then feed to the CNN parameter predictor. The CNN parameter predictor then tries to predict the parameters of DIP. Following that, the image filtered by the DIP module is fed as input to the YOLOv3 [24] detector for the pedestrian detection task.

CNN-Based Parameter Predictor. As shown in Fig. 1, the CNN parameter predictor module consists of five convolutional blocks followed by two fully-connected layers. Each convolutional block contains a 3×3 convolutional layer with stride 2 and a leaky ReLU activation layer. The module tries to understand the global content of the image, such as brightness, color and tone and the degree of fog in order to predict the parameters required by the DIP module. In order to save computation cost, the input images are downsampled to a lower resolution of 256×256 using bilinear interpolation. The output channels of the five convolutional layers are 16, 32, 32, 32 and 32, respectively. The output of this module is fed into the DIP module.

DIP Module. The DIP module consists of six differentiable filters with adjustable hyperparameters, namely Defog, White Balance (WB), Gamma, Contrast, Tone and Sharpen. According to Hu et al. [13], the standard color and tone operators, such as White Balance, Gamma, Contrast and Tone, can be expressed as pixel-wise filters. Therefore, the filters can be classified into three categories namely, Pixel-wise, Defog and Sharpen Filters. The Defog filter has been designed for foggy scenes only.

Pixel-Wise Filters. In pixel-wise filters, an input pixel value $P_i = (r_i, g_i, b_i)$ is mapped into an output pixel value $P_o = (r_o, g_o, b_o)$ where (r, g, b) represent the values of the red, green and blue color channels, respectively. The mapping functions of the pixel-wise filters have been shown in Table 1.

Table 1. Mapping functions of pixel wise filters.

Filter	Parameters	Mapping Function
Gamma	G: gamma value	$P_o = P_i^G$
WB	W_r, W_g, W_b: factors	$P_o = (W_r.r_i, W_g.g_i, W_b.b_i)$
Tone	t_i: tone params	$P_o = (L_{t_r}(r_i), L_{t_g}(g_i), L_{t_b}(b_i))$
Filter	α: contrast value	$P_o = \alpha.En(P_i) + (1 - \alpha).P_i$

Defog Filter. We used a defog filter designed using the *dark channel prior method* by He *et al.* [11]. The formation of a hazy image can be formulated as follows:

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{1}$$

where $I(x)$ is the foggy image, $J(x)$ represents the scene radiance (clean image), A is the global atmospheric light, and $t(x)$ is the medium transmission map. The atmospheric light A and the transmission map $t(x)$ need to be obtained to recover the clean image $J(x)$. At first, the dark channel map of the haze image $I(x)$ has been computed and the top 1000 brightest pixels have been picked. Then, the average of those 1000 pixels of the corresponding position of the haze image $I(x)$ has been taken to estimate the value of A.

Sharpen Filter. We use the sharpen filter to enhance the image details. For sharpening the images, the following equation describes the process:

$$F(x, \lambda) = I(x) + \lambda(I(x) - Gau(I(x))) \tag{2}$$

where $I(x)$ is the input image, $Gau(I(x))$ denotes Gaussian filter, and λ is a positive scaling factor.

Detection Module. We use the single-stage YOLOv3 detection network, with the same network architecture and loss function as the original YOLOv3 [24]. YOLOv3 contains darknet-53 which has successive 3×3 and 1×1 convolutional layers based on the idea of ResNet.

2.2 Timed Quality Temporal Logic (TQTL)

Timed Quality Temporal Logic (TQTL) [6,12] is an extension of Timed Propositional Temporal Logic (TPTL) which incorporates syntax and semantics for reasoning about data from perception systems specifically. The syntax defines operators to reason about classes of detected objects and the confidence associated with the detection outputted by perception systems.

TQTL Syntax. A TQTL formula φ over a finite set of predicates $\mathcal{P}$, a finite set of frame number variables (ν_t), and a finite set of object ID variables (ν_{i_d}) can be defined according to the following grammar:

$$\begin{aligned} \varphi ::= & \top \mid \mu \mid t.\varphi \mid \exists id@t, \varphi \mid \forall id@t, \varphi \mid \\ & t \leq u + n \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \cup \varphi_2 \end{aligned} \tag{3}$$

The time constraint is $t \leq u + n$, which implies the timespan starting from u and spanning across n consecutive frames.

TQTL Semantics. The semantics of TQTL maps a data stream $\mathcal{D}$, which is a sequence of video frames containing multiple candidate objects in each frame, a frame number $\top$ and a valuation function ν to a real-valued entity. A valuation function in this context is a function that assigns some values to frames and objects present in the corresponding frames. TQTL mainly deals with the task that if a particular object is tracked across multiple frames of a video, the probability of detecting it does not fall below a certain threshold across a certain number of consecutive frames. The function $[\![\cdot]\!]$ can be defined recursively as follows:

$$
\begin{aligned}
[\![\top]\!](\mathcal{D},\tau,\nu) &= +\infty \\
[\![\mu]\!](\mathcal{D},\tau,\nu) &= \theta(\nu(f_\mu(t_1,...,t_{n_1},id_1,...,id_{n_2})),c) \\
[\![t.\varphi]\!](\mathcal{D},\tau,\nu) &= [\![\varphi]\!](\mathcal{D},\tau,\nu[t \leftarrow \tau]) \\
[\![\exists id@t,\varphi]\!](\mathcal{D},\tau,\nu) &= \max_{k \epsilon \mathcal{S}(\mathcal{D}_{v(t)})} [\![\varphi]\!](\mathcal{D},\tau,\nu[id \leftarrow k]) \\
[\![t \leq u+n]\!](\mathcal{D},\tau,\nu) &= \begin{cases} +\infty, & \text{if } \nu(t) \leq \nu(u)+n, \\ -\infty, & \text{otherwise.} \end{cases} \\
[\![\neg\varphi]\!](\mathcal{D},\tau,\nu) &= -[\![\varphi]\!](\mathcal{D},\tau,\nu) \\
[\![\varphi_1 \wedge \varphi_2]\!](\mathcal{D},\tau,\nu) &= \min([\![\varphi_1]\!],[\![\varphi_2]\!](\mathcal{D},\tau,\nu)) \\
[\![\varphi_1 \cup \varphi_2]\!](\mathcal{D},\tau,\nu) &= \max_{\tau' \geq \tau} \min \begin{pmatrix} [\![\varphi_2]\!](\mathcal{D},\tau',\nu), \\ \min_{\tau'' \in [\tau,\tau')} [\![\varphi_1]\!](\mathcal{D},\tau'',\nu) \end{pmatrix}
\end{aligned}
$$

3 Our Approach

The pipeline consists of the IA-YOLO model for processing the input frame and detecting pedestrians along with a TQTL monitor for verifying a given specification under adverse weather conditions. The CNN parameter predictor module of the IA-YOLO takes an input frame and outputs the parameters for the different filters of the DIP module which then generates an enhanced image containing the latent information. In the second step, a TQTL monitor for a given specification φ monitors the output of the IA-YOLO model, reporting whether the specification has been satisfied or violated (Fig. 2).

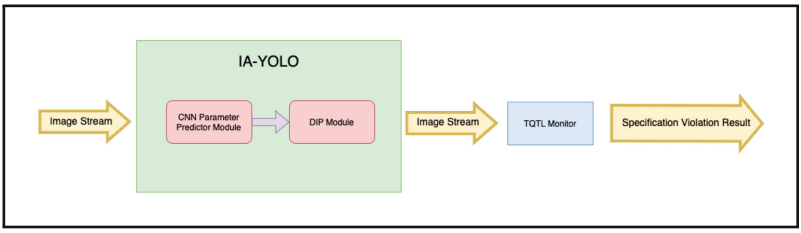


Fig. 2. TQTL Monitoring Pipeline for IA-YOLO

4 Experimental Analysis

In our case, we focus on detecting pedestrians in foggy and night conditions. We aim to validate the following specification: "If a person is detected with a confidence score greater than or equal to 0.3 in a particular frame, then in the next 4 frames, the probability of detecting the same person should never drop below 0.25." This specification is represented by the following TQTL expression:

$$\begin{aligned}\varphi = \square(x.\forall id_1 @ x, (C(x, id_1) = Pedestrian \wedge P(x, id_1) \geq 0.3) \\ \rightarrow \square(y.((x \leq y \wedge y \leq x + 4) \\ \rightarrow C(y, id_1) = Pedestrian \wedge P(y, id_1) > 0.25))\end{aligned}$$

We have chosen the confidence scores to be 0.3 and 0.25 respectively to account for the comparatively poorer performance of detection models in adverse conditions as compared to clear weather conditions and to aid the comparison task between IA-YOLO and vanilla YOLO.

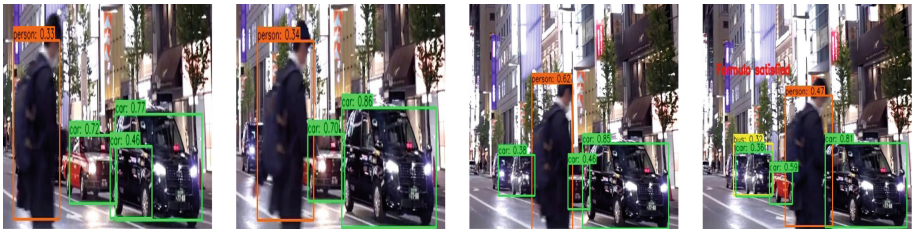
For the purpose of our experimentation, we have created a custom evaluation dataset containing night and foggy driving videos from dash-cam driving videos that are publically available on YouTube. In order to create our dataset, we have taken small clips from these videos where a pedestrian is found be crossing the road front of the car. Note that while these video clips do not contain any ground truth annotations, our evaluation metric is based on the quantitative semantics of TQTL, which do not rely on groundtruth.

To evaluate the models, we compute the robustness of both vanilla YOLO and IA-YOLO on the videos contained the dataset, with respect to the above specification φ. The results are shown in Table 2, and Fig. 3 and Fig. 4 show some examples of sequences of images that satisfy or violate the specification φ. We

Table 2. Robustness values achieved on custom dataset using Vanilla YOLO and IA-YOLO models against φ.

Weather Condition	Robustness φ	
	Vanilla YOLO	IA-YOLO
Foggy	−0.25	0.31
	−0.25	0.36
	−0.25	0.29
	−0.25	0.51
Night	0.28	0.54
	0.10	0.22
	−0.25	0.22
	0.07	0.3
	0.29	0.58
	−0.25	0.18

(a) vanilla YOLO: Pedestrian detected and TQTL formula satisfied.



(b) IA-YOLO: Pedestrian detected and TQTL formula satisfied.

Fig. 3. Monitoring results in night conditions.

(a) vanilla YOLO: Pedestrian detected but TQTL formula violated.

(b) IA-YOLO: Pedestrian detected and TQTL formula satisfied.

Fig. 4. Monitoring results in foggy conditions.

find that vanilla YOLO does not sufficiently satisfy the TQTL specification in foggy conditions, but detects pedestrians reasonably well night-time conditions. On the other hand, the IA-YOLO model is found to be able to detect pedestrians succesfully in both foggy and night conditions.

5 Conclusion

The IA-YOLO model has been able to detect pedestrians satisfactorily in foggy and night conditions. IA-YOLO performs better as compared to vanilla YOLO in adverse conditions. However, we found some cases where it fails to detect pedestrians in one of the consecutive frames with the desired level of confidence. The robustness of this model has been estimated using TQTL which has been able to correctly verify the required specification. This quality metric will help in debugging or improving the existing model and lead to better detection results in foggy and night conditions in a safety-critical context.

We plan to develop a more robust YOLO model for object detection in low-light conditions by fine-tuning the current model. We also aim to reduce the runtime of the entire monitoring process by only applying the DIP module on the frames where the TQTL monitor fails instead of all the frames. Moreover, we hope to create a training dataset which will be used to train the CNN-PP module more efficiently with the images which are flagged as "violated" by the TQTL monitor.

Acknowledgement. The authors would like to thank the anonymous reviewers for their feedback. This work was supported by the National Science Foundation through the following grants: CAREER award (SHF-2048094), CNS-1932620, CNS-2039087, FMitF-1837131, CCF-SHF-1932620, the Airbus Institute for Engineering Research, and funding by Toyota R&D and Siemens Corporate Research through the USC Center for Autonomy and AI.

References

1. Autonomous Vehicle Collision Reports. Technical report, California Department of Motor Vehicles (2023). www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports/
2. Antonante, P., Spivak, D.I., Carlone, L.: Monitoring and Diagnosability of Perception Systems. arXiv:2005.11816 [cs] (2020)
3. Balakrishnan, A., Deshmukh, J., Hoxha, B., Yamaguchi, T., Fainekos, G.: PerceMon: online monitoring for perception systems. In: Feng, L., Fisman, D. (eds.) RV 2021. LNCS, vol. 12974, pp. 297–308. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88494-9_18
4. Balakrishnan, A., et al.: Specifying and evaluating quality metrics for vision-based perception systems. In: 2019 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1433–1438 (2019). https://doi.org/10.23919/DATE.2019.8715114
5. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
6. Dokhanchi, A., Amor, H.B., Deshmukh, J.V., Fainekos, G.: Evaluating perception systems for autonomous vehicles using quality temporal logic. In: Colombo, C., Leucker, M. (eds.) RV 2018. LNCS, vol. 11237, pp. 409–416. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03769-7_23

7. Dokhanchi, A., Hoxha, B., Tuncali, C.E., Fainekos, G.: An efficient algorithm for monitoring practical TPTL specifications. In: 2016 ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE), pp. 184–193. IEEE (2016)
8. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vision **111**(1), 98–136 (2015). https://doi.org/10.1007/s11263-014-0733-5
9. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
11. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2010)
12. Hekmatnejad, M.: Formalizing Safety, Perception, and Mission Requirements for Testing and Planning in Autonomous Vehicles. Ph.D. thesis, Arizona State University (2021)
13. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: a white-box photo post-processing framework. ACM Trans. Graph. (TOG) **37**(2), 1–17 (2018)
14. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
16. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
17. Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., Zhang, L.: Image-adaptive YOLO for object detection in adverse weather conditions. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
18. Michaelis, C., et al.: Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming (2020). arXiv:1907.07484 [cs, stat]
19. Narasimhan, S.G., Nayar, S.K.: Chromatic framework for vision in bad weather. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662), vol. 1, pp. 598–605. IEEE (2000)
20. Padilla, R., Netto, S.L., da Silva, E.A.B.: A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237–242 (2020). https://doi.org/10.1109/IWSSIP48289.2020.9145130, iSSN: 2157-8702
21. Padilla, R., Passos, W.L., Dias, T.L.B., Netto, S.L., da Silva, E.A.B.: A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics **10**(3), 279 (2021). https://doi.org/10.3390/electronics10030279
22. Qin, Q., Chang, K., Huang, M., Li, G.: DENet: detection-driven enhancement network for object detection under adverse weather conditions. In: Proceedings of the Asian Conference on Computer Vision, pp. 2813–2829 (2022)
23. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
24. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

25. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
26. Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K.: Instant dehazing of images using polarization. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–I. IEEE (2001)
27. Teeti, I., Musat, V., Khan, S., Rast, A., Cuzzolin, F., Bradley, A.: Vision in adverse weather: Augmentation using CycleGANs with various object detectors for robust perception in autonomous racing (2023). arXiv:2201.03246 [cs]
28. Wu, B., Iandola, F., Jin, P.H., Keutzer, K.: SqueezeDet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 129–137 (2017)
29. Xu, H., Gao, Y., Yu, F., Darrell, T.: End-to-end learning of driving models from large-scale video datasets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2174–2182 (2017)
30. Xu, Y., Weaver, J.B., Healy, D.M., Lu, J.: Wavelet transform domain filters: a spatially selective noise filtration technique. IEEE Trans. Image Process. **3**(6), 747–758 (1994)