



Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles

Liang Peng¹ · Hong Wang¹ · Jun Li¹

Received: 4 January 2021 / Accepted: 31 May 2021 / Published online: 30 July 2021
© The Author(s) 2021

Abstract

The safety of the intended functionality (SOTIF) has become one of the hottest topics in the field of autonomous driving. However, no testing and evaluating system for SOTIF performance has been proposed yet. Therefore, this paper proposes a framework based on the advanced You Only Look Once (YOLO) algorithm and the mean Average Precision (mAP) method to evaluate the object detection performance of the camera under SOTIF-related scenarios. First, a dataset is established, which contains road images with extreme weather and adverse lighting conditions. Second, the Monte Carlo dropout (MCD) method is used to analyze the uncertainty of the algorithm and draw the uncertainty region of the predicted bounding box. Then, the confidence of the algorithm is calibrated based on uncertainty results so that the average confidence after calibration can better reflect the real accuracy. The uncertainty results and the calibrated confidence are proposed to be used for online risk identification. Finally, the confusion matrix is extended according to the several possible mistakes that the object detection algorithm may make, and then the mAP is calculated as an index for offline evaluation and comparison. This paper offers suggestions to apply the MCD method to complex object detection algorithms and to find the relationship between the uncertainty and the confidence of the algorithm. The experimental results verified by specific SOTIF scenarios proof the feasibility and effectiveness of the proposed uncertainty acquisition approach for object detection algorithm, which provides potential practical implementation chance to address perceptual related SOTIF risk for autonomous vehicles.

Keywords SOTIF · Uncertainty evaluation · Confidence calibration · Autonomous vehicles

Abbreviations

AP	Average precision
COCO	Common objects in context
HB	Histogram binning
IoU	Intersection over union
mAP	Mean average precision
MCD	Monte Carlo dropout
mmAP	Mean mean average precision
SOTIF	The safety of the intended functionality
YOLO	You only look once

1 Introduction

According to the data provided by World Health Organization, about 1.25 million people are killed and even more people are injured in traffic accidents each year. A significant amount of work has been carried out along with technological advancements to ensure the safety of automobiles. For example, the advanced driving assistance system (ADAS) plays an auxiliary role for the general public to drive, and the intelligent transportation system (ITS) also brings convenience to road safety management. However, developing a more powerful automatic driving system and evaluating its safety are still great challenges that need further studies.

The safety concerns of automobiles mainly include functional safety, automotive cybersecurity, and safety of the intended functionality (SOTIF), which focuses on the risks caused by potential hazards such as functional insufficiencies and reasonably predictable personnel misuses [1]. The SOTIF performance of the perception algorithms is mainly considered in this research, to evaluate which a relevant database is needed as support. Establishing the SOTIF-related

✉ Hong Wang
hong_wang@tsinghua.edu.cn

¹ Tsinghua University, Beijing, China

scenario database is becoming a hot field, and to the best of the authors' knowledge, there are no open resources yet. This paper establishes such a scenario database and the images are applied to the tests.

There are quite a few effective evaluation indicators in computer vision, and the performance of deep learning models can be evaluated from different aspects. The metrics include confusion matrix, precision-recall (P-R) curve, receiver operating characteristic (ROC) curve, and F-score, all of which have been widely used in the evaluation of image classification models [2]. However, the outputs of the object detection models are unstructured and have strong uncertainty, which not only involves a variety of object categories but also needs accurate positioning. As a result, the evaluation method is more complex than that of the image classification task. A commonly used performance evaluation system in object detection is mean average precision (mAP). In the mAP evaluation system, all the prediction boxes of the same category are sorted according to their confidence levels. Then, the intersection over union (IoU) of the prediction and the related ground truth bounding boxes is used to express the positioning accuracy to distinguish the positive and the negative samples. After that, average precision (AP) is calculated by drawing the P-R curves, mAP is calculated by averaging the APs among different object categories, and the mean mean average precision (mmAP) metric is obtained by averaging the mAPs among different IoU thresholds.

The mAP evaluation system, despite being widely used, has its limitations. This evaluation system depends on the subjective confidence level of the output of the perception model. However, there is sometimes a large gap between the original confidence level and the actual accuracy, which often leads to the situation of diffidence or overconfidence. Therefore, it can obtain more reliable evaluation results by calibrating the confidence level. The process of confidence calibration has been extensively studied in the image classification task [3]. Histogram Binning (HB) proposed by Zadrozny et al. and Isotonic Regression (IR) proposed by Elkan et al. do not require modifying the training parameters of the neural network [4, 5], while Bayesian Binning into Quantiles (BBQ) proposed by Naeini et al. and Platt scaling method proposed by Platt et al. need to modify the loss function and train more parameters to calibrate the outputs [6, 7]. However, due to the complex and uncertain output, little work has been done to calibrate the confidence level in the object detection task.

Theoretically, the more uncertain a model's output is, the lower the confidence level should be. Therefore, this paper attempts to calibrate the confidence level using the model uncertainty. However, before calibration, the uncertainty of a model needs to be captured first. Through the Bayesian theorem and a series of simplification and derivation, the

problem of solving model uncertainty can be transformed into the problem of calculating the posterior distribution of model weights. However, due to the nonlinearity and the non-conjugation of the neural network structure, it is difficult to achieve accurate posterior reasoning. In practical applications, researchers have adopted various approximation methods to carry out Bayesian inference [8, 9]. Hamiltonian Monte Carlo (HMC) was explored by defining an invariant distribution as the posterior distribution of a Markov chain [10]. Mean field variational inference attempted to find a Gaussian distribution to approximate the posterior distribution of the model [11]. Bootstrap resampled the training data to generate multiple datasets and used them to obtain different models [12]. Several other approximate reasoning methods are available for the Bayesian neural network, including stochastic search and probabilistic backpropagation [13]. Most of these methods are based on variational inference and can optimize the variational lower bound. Moreover, the deep learning models need to be modified and trained again, and the loss function must be adjusted according to different optimization methods.

Monte Carlo dropout (MCD) proposed by Gal.Y et al. is an approximate variational reasoning method based on dropout [14]. The approximate distribution is the product of Bernoulli variables and corresponding model weights, and the only required parameter is the dropout rate, p , of Bernoulli distribution [14]. Zhu et al. further found that the result of uncertainty estimation is robust when p is within a reasonable range [13]. Therefore, the MCD method does not need to change the original network structure and the parameters of the existing model, and it can be directly applied to the previously trained model. It has the advantages of simplicity, universality, and easy implementation. Therefore, this study chooses the MCD method to capture the model uncertainty.

This paper presents an evaluation system for object detection models based on model uncertainty analysis and confidence calibration. Based on the SOTIF-related scenarios collected by the authors, this paper uses the MCD method to quantify the uncertainty of the excellent one-step object detection model YOLOv3 [15] and further calibrates the output confidence level. The validity of the confidence calibration method is verified through experiments, which improves the reliability of the mAP evaluation results. To the best of the authors' knowledge, using model uncertainty in confidence calibration has not yet been fully explored in the field. When the perceptual algorithm outputs its uncertainty, the reliability of its confidence is improved, thus the subsequent systems of autonomous vehicles will make safer decisions [16, 17, 33].

The remainder of the paper is organized as follows. Section 2 presents the design of the evaluation system, including the overall framework, the implementation of uncertainty analysis, the calculation of confidence calibration factors,

and the evaluating method. Section 3 presents the experiments and discusses the results, including establishing a small SOTIF dataset and its application in this study to validate the confidence calibration method and the reliability of the evaluation index. Lastly, conclusions and some future work are presented in Sect. 4.

2 Evaluation System

This section introduces the improved mAP evaluation system of object detection algorithms, including the construction of the framework, the implementation of uncertainty analysis, the design of confidence calibration factors, and the evaluating method.

2.1 Overall Framework of the Evaluation System

Figure 1 shows the overall framework of the designed evaluation system for object detection algorithms which includes the uncertainty analysis module, the calibration module and the evaluation module. The uncertainty analysis module receives the SOTIF dataset as the input, calculates the prediction uncertainty using the MCD method, and outputs them to the calibration module. Then, the calibrated results are sent to the evaluation module to finally evaluate the algorithm. The three main modules are illustrated below, and more information about the SOTIF dataset that includes fog, snow, glare, and rain subsets is introduced in Sect. 3.

In the uncertainty analysis module, the original YOLOv3 model is sampled N times through random dropout to change the weights. Therefore, different detection results $y_{(N)}^*$ can be obtained from the same test input x^* . Following is an IoU

comparator that is used to match the bounding boxes of the same object in different sampled detection results. The IoU comparator generates a cluster for every object in the first sampled detection result, and then a subsequent bounding box will be assigned to a cluster only if the IoU of them exceeds the given IoU threshold. Then, the variance calculated with the bounding boxes in the same cluster can be used to represent the epistemic uncertainty accordingly. Meanwhile, the mean square error between the ground truth $\{y'_v\}$ of the validation set $\{x'_v\}$ and the detection results through the original model $\{y'_v\}$ is regarded as the aleatoric uncertainty. Then, the prediction uncertainty of the model can be obtained by adding the two uncertainties. In the calibration module, some calibration factors are designed, and the information from the uncertainty analysis module is used to calibrate the confidence of the model. After obtaining the calibrated result, the evaluation indexes at all levels are calculated. In the evaluation module, all the detected boxes are classified as true positive (TP), false negative (FN), true negative (TN), or false positive (FP) considering their detection results compared with the ground truth and their uncertainties. Finally, the metrics such as precision, recall, and AP are calculated to evaluate the algorithm performance in SOTIF-related scenarios.

2.2 Quantification of Model Uncertainty

There are uncertainties in the convolution neural network that are difficult to be explained by human beings. In the field of uncertainty analysis of deep learning models, Bayesian inference is the current mainstream method [18, 19]. According to Bayes theorem, it can be concluded that

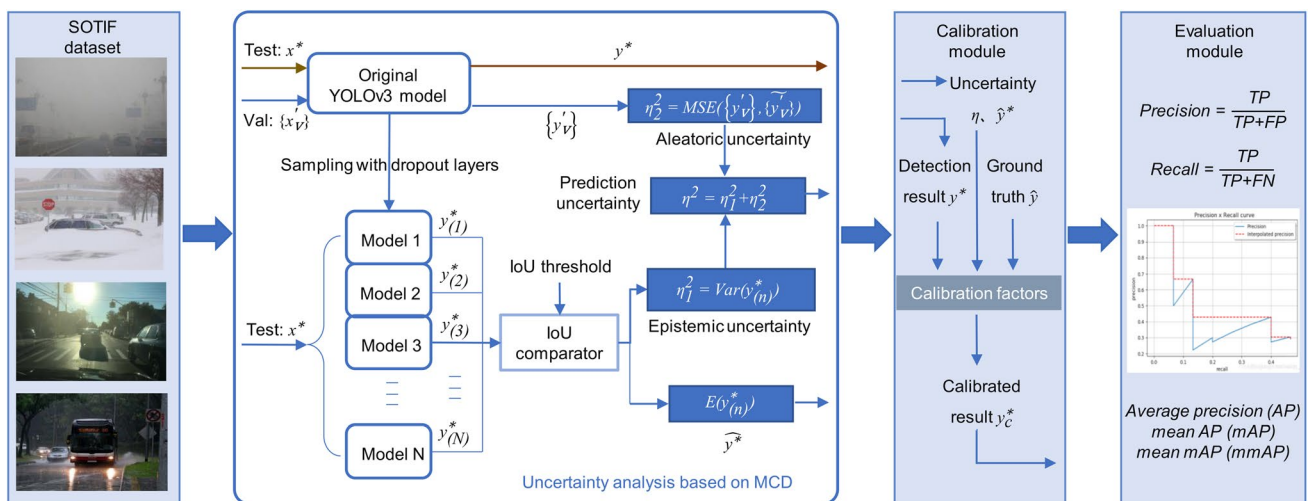


Fig. 1 Overall framework of the mAP evaluation system of the YOLOv3 object detection algorithm considering uncertainty

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, f)p(f|X, Y)df \tag{1}$$

where X and Y are the training dataset and the labeled ground truth, respectively, x^* is a new input data point, y^* is the corresponding output result, and f represents the model. $p(y^*|x^*, X, Y)$ describes the distribution of output y^* of an unknown model trained from X and Y and represents the uncertainty of model output. $p(y^*|x^*, f)$ describes the distribution of output y^* of a given model f , which can be estimated directly. $p(f|X, Y)$ describes the distribution of model f trained by given training set X and Y , i.e., the posterior distribution of model f . The concept of model f is quite abstract and can be simplified by the limited weight parameter W :

$$p(y^*|x^*, X, Y) = \int_W p(y^*|x^*, W)p(W|X, Y)dW \tag{2}$$

Therefore, the problem of obtaining model uncertainty is transformed into the problem of estimating the posterior distribution of model weights using the above formula. Then, the MCD method is used to quantify the model uncertainty:

$$W = M * \text{diag}\left(\left[z_{i,j}\right]_{j=1}^{K_i}\right), z_{i,j} B(1, p_i) \tag{3}$$

The core of the MCD method is to use multiple Bernoulli distribution $q(W)$ to approximate the posterior distribution $p(W|X, Y)$, where M is the weight matrix of the original model, $z_{i,j}$ represents the j th input neuron in layer i of the model that obeys the Bernoulli distribution with probability of p_i . The zero value of $z_{i,j}$ indicates that the corresponding neuron is inactivated. After this processing, a new weight matrix of the model is produced. For a given input, different outputs will be obtained. It should be noted that the adoption of dropout will change the expectation of model output. For example, a neuron node that originally outputs 1 now outputs 1 with probability p_i and 0 with probability $1 - p_i$. Therefore, the output of each neuron must be divided by p_i to maintain the original expectation.

Another key point of the MCD method is to use the Monte Carlo method to obtain a series of sampling models. The original model weight is sampled N times, and N y^* output results are obtained. The uncertainty of the model can be approximated by the variance of the samples:

$$\text{Var}(y_{(n)}^*) = \frac{1}{N} \sum_{n=1}^N (y_{(n)}^* - \bar{y}^*)^T (y_{(n)}^* - \bar{y}^*) \tag{4}$$

Herein, the model YOLO-416 pre-trained by Microsoft Common Objects in Context (MS COCO 2017 train) is used and the number of model samples is set to 20 [20]. The YOLO-416 model is equipped with the YOLOv3 algorithm and reorganizes the input into 416×416 . Since YOLOv3 has removed the dropout layer and replaced it with batch

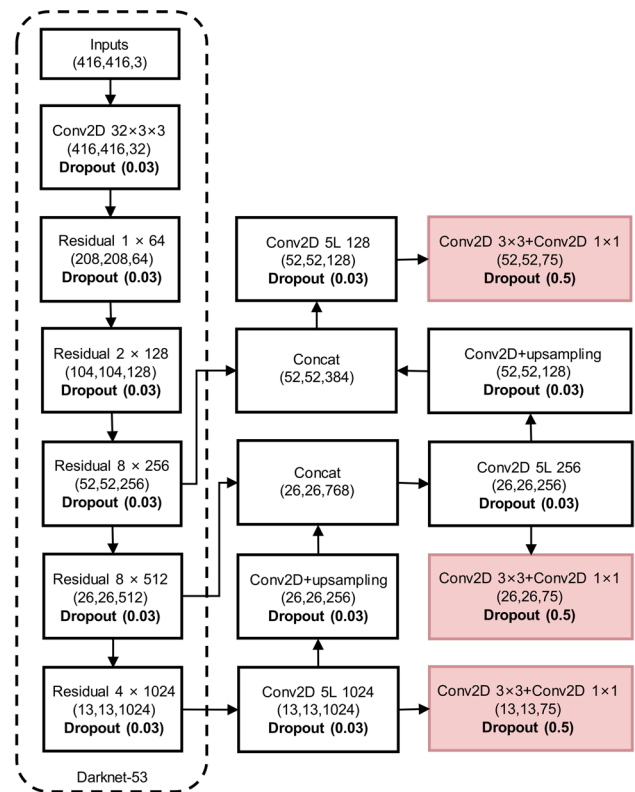


Fig. 2 Modified network architecture of YOLO-416

normalization, the network architecture of YOLO-416 is manually modified based on the MCD method, as shown in Fig. 2. Another related work [34] also reimplemented YOLOv3 to estimate model uncertainty through MCD method. However, they only introduced dropout layers towards the end of the network and used the deterministic feature tensor to improve the efficiency, while we add dropout layers after each convolution module to simulate the complete model uncertainty.

According to Ref. [21], while using dropout as the regularization method, the dropout rate of 0.5 for the full connection layer has been used to randomly generate more network structures and solve the problem of overfitting. Meanwhile, the 1×1 convolution layer at the end of the prediction network has similar effects as the full connection layer and more parameters than other convolution layers. Therefore, the dropout rate $1 - p_i$ is set to 0.5 for the 1×1 convolution layer. However, for the other convolution layers, as the convolutional shared-filter architecture brings a drastic reduction in the number of parameters, the large dropout rate will lead to a severe loss of features. Through testing on a few images, the dropout rate is set to 0.03 for the other convolution layers. Moreover, these selected rates are hyperparameters and can be fine-tuned further.

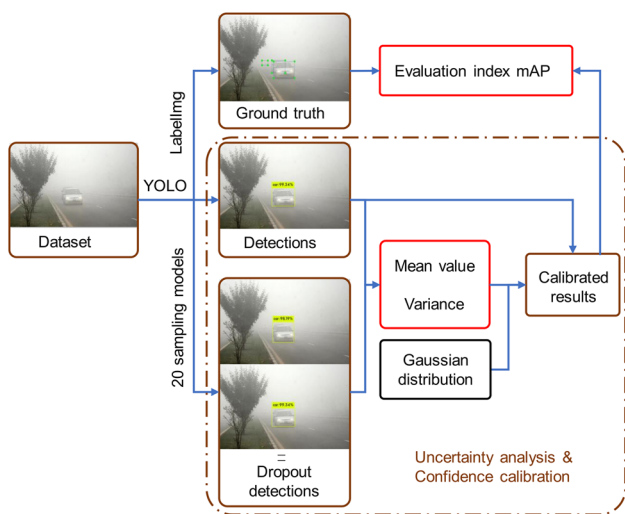


Fig. 3 Data flow in confidence calibration module

2.3 Calibration of Output Confidence

As mentioned above, HB is a simple nonparametric calibration method. It divides all uncalibrated possible confidence values c_i into mutually exclusive m bins, and each bin specifies a calibration score c_m . If the output falls into the n th bin, the calibration result will be c_n .

This paper uses the results from uncertainty analysis to calibrate the confidence. From the uncertainty analysis module, the mean and the variance of the sampling detection results of each data point under different sampling models can be obtained. Then, the distribution of the prediction result of the original model relative to the mean value can be obtained, i.e., the original prediction result is within k times the standard deviation of the mean. Therefore, the number k can be used to divide the bins.

The calibration factors are selected according to the following rules: The closer the original test result is to the sampling mean value, the more stable the model output is, and the larger the calibration factor is, and vice versa. While the output is stable enough and the model is confident itself, an upper confidence limit after calibration should be set to 1. Furthermore, the expectations before and after the calibration should be kept unchanged to reduce other impacts. In the application research of uncertainty analysis, Gal and Zhu et al. all mentioned that the outputs of visual tasks could be considered to obey a priori normal distribution [22]. Therefore, the calibration factors are designed based on the probability distribution of normal distribution, as shown in Fig. 3 and Table 1.

In the tests, it is observed that the object detected by the original model may not be detected in the sampled models. While designing the algorithm, the M ($M \leq N$) detected sampling results of a certain object are used to calculate its

Table 1 Design of calibration factors

k	Probability	Calibration factor
$0 < k \leq 1$	0.6827	1.05
$1 < k \leq 2$	0.2718	0.9
$2 < k \leq 3$	0.0428	0.85
$k > 3$	0.0027	0.8
Expectation	1	0.999995

uncertainty. In a certain scenario, an object is only detected twice in 20 sampling models, which indicates high model uncertainty. However, both results are very close, which leads to the statistical value underestimating its uncertainty. Therefore, the factor of M is also considered in the calibration. The lower M is, the lower the output confidence should be. At the same time, the object that is not detected by the original model may also be detected by the sampling models. Since an uncertainty range is added to the existing output of the original model, these cases are ignored. In several studies, the MCD method has been integrated into the model to form a Bayesian network, using the mean value as the model output [23, 24]. This kind of method is suitable for semantic segmentation, but not for one-step object detection. Because the former is essentially the classification of pixels, while the latter is a regression problem having difficulties in matching the bounding boxes detected by different sampling models. Miller studied some methods of clustering the bounding boxes based on spatial and semantic affinity and found that a basic sequential algorithmic scheme (BSAS) method with the IoU affinity measure did well [25, 26]. This paper follows Miller’s work to generate detection clusters for further calculating [27].

Finally, some functions and scripts are written in MATLAB to realize the confidence calibration method. After inputting the ground truth, the original prediction results, the 20 sampling detection results, and the final results after calibration are obtained. The data flow is shown in Fig. 3. The confidence calibration formula is

$$y_c^* = \frac{M}{N} \times k \times y^* \tag{5}$$

2.4 Evaluation Method for Object Detection

As mentioned in Sect. 1, the mAP evaluation system is a benchmark method to evaluate object detection algorithms but needs further improvement. This paper uses the mAP evaluation system as the overall evaluation framework while performing some optimization considering the characteristics of the object detection task. For one thing, the credibility of the evaluation results is improved by calibrating the confidence. For another, the confusion matrix from the

classification task is extended to determine TP, FN, TN, and FP.

The confusion matrix is one of the most basic and intuitive methods to measure a classification model. It takes the number of samples predicted by the model as the column, and the number of samples with the ground truth label as the row to form a matrix. Then, the numbers on the diagonal represent the samples whose prediction results are consistent with the real results, and the larger the numbers are, the better the prediction results are [28]. For the image classification task, each image has a certain output and the result is either correct or wrong. However, the object detection task has two additional situations, i.e., repetition prediction and missing prediction. The repetition prediction means that no less than two bounding boxes are generated for an object, or bounding boxes are generated for non-existent objects. The missing prediction means that the corresponding bounding box is not generated for an actual object. To take these cases into account, the confusion matrix is extended by adding an extra column and an extra row to record the missing predictions and the repetition predictions, respectively.

The extended confusion matrix can retain the original confusion matrix in the top left corner to reflect the classification effect of the algorithm. It can also record the repetition and missing situations, and evaluate the comprehensive performance of the model. The submatrix consisting of the first four rows and columns in Fig. 4 is a diagonal matrix, indicating that the model does well in classification and will not classify a cyclist into a car. The most challenging problem for object detection is reducing missing predictions. At the same time, the method to obtain indicators like TN is also extended, as shown in Fig. 4. No wrong detections occur in the Snow subset, indicating that the classification ability of the model is good. However, there are some samples of repetition predictions and missing predictions. For the pedestrian category, five objects are missing, meaning five false negatives. Therefore, 5 should be added to the FN.

3 Experiments and Test Results

In this section, firstly the established database is introduced, and the SOTIF dataset is generated. Then, a set of similar scenarios are selected to observe the results of the uncertainty analysis. Next, the proposed evaluation system of object detection algorithms is applied to the SOTIF dataset and YOLOv3 algorithm. The effectiveness of the optimization scheme is verified and the SOTIF performance of the YOLO-416 model is reliably evaluated. Finally, some other SOTIF-related scenarios are extracted from the latest database to explore how the uncertainty and safety information can be passed down to the decision layer.

	Pedestrian	Cyclist	Car	Traffic signal	Missing	Predictions
Pedestrian	TP: 34					FN: 5
Cyclist		4				1
Car			104			23
Traffic signal				20		2
Repetition			8	1		
Ground truth						

Fig. 4 Extended confusion matrix of results of Snow subset

3.1 The SOTIF Dataset

In the field of computer vision, many well-known datasets and competition platforms are available for algorithm developers to test and compare the performance of their networks. However, such datasets usually contain quite a few indoor and non-traffic outdoor scenes. Therefore, they are too broad and unrepresentative for autonomous vehicles. Several datasets have also been created for autonomous driving. The KITTI dataset introduced by Geiger et al. is available for optical flow and object detection [29]. The data has been captured by the autonomous driving platform Anniway, but mostly in clear weather. Yu et al. contributed the BDD100k dataset comprising of over 100 K driving videos and having diversity in geography, environment, and weather [30].

As the SOTIF has gained incremental attention, new requirements for testing the scenarios have appeared. The authors' team is establishing a database of scenarios related to the SOTIF, focusing on traffic scenarios affected by various factors such as light intensity and weather. It is believed that the scenario-based test technology provides an effective means for the test and the evaluation of the SOTIF of the intelligent driving system. The overall train of thought of establishing the database follows the standard ISO 21,448, considering the potential hazards from insufficient functions and personnel misuse. By analyzing the functional limitations and the defects of the state-of-the-art solutions of perception, localization, human-machine interface (HMI), decision-making, and control systems in the intelligent driving system, the test requirements for SOTIF in those aspects are proposed. Then, a seven-layer SOTIF test scenario framework is designed for intelligent driving vehicles, including information of road structure, traffic facilities, temporary changes of roads and facilities, traffic participants, climate and environment, wireless communication, and ego status. So far, the preliminary SOTIF test scenario database based on the proposed test requirements has been established. Currently, a number of annotated images and videos are included in the database, and the plan is to generate simulation scenarios that meet the specifications like OpenSCENARIO [31].

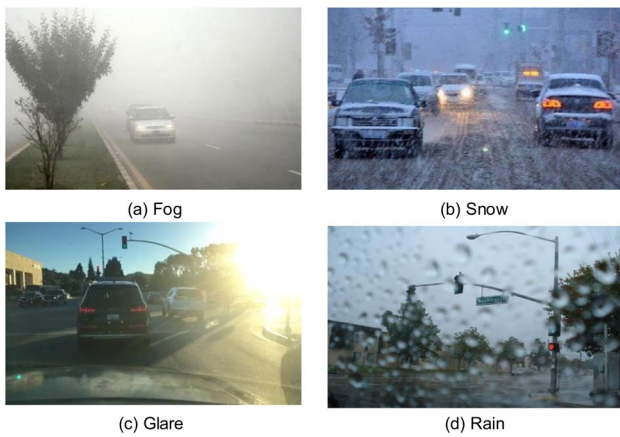


Fig. 5 Example images of SOTIF dataset

In the perception layer, the functional limitations of the sensors and the interferences from the external environment are analyzed. The camera, LiDAR, Millimeter-Wave Radar (MMWR), ultrasonic radar, and the perception fusion of these sensors are mainly considered. For the camera, the perceptions under extreme weather such as rain, snow, fog, and changing and adverse lighting conditions such as glare are considered. From the above-mentioned database, some images are intercepted to form a SOTIF dataset for application in this study. Some of these images are selected from well-known datasets such as COCO, KITTI, and Apolloscape, some are obtained from search engines like Baidu and Google, and some are captured by the authors and their partners. The SOTIF dataset used in this study is divided into four subsets: Fog (22 images), Snow (32 images), Glare (103 images), and Rain (140 images). They cover the scenarios in which these natural environmental factors affect the SOTIF of the perception algorithms. Figure 5 shows some examples.

Herein, the software LabelImg is used to manually draw the bounding box of every object for each image as the ground truth. Since the sample size is small, the object classification is simplified. The objects are mainly divided into four categories that are the most common in road traffic scenes, i.e., pedestrian, cyclist, car, and traffic signal. Take the car category as an example, the cars, trucks, and buses of the original COCO outputs are all included in it. A total of 2045 bounding boxes are drawn for 297 images. The detailed data is recorded in Table 2.

3.2 Case study of Uncertainty Analysis

To study the influence of the environment on the uncertainty of the model output, three similar scenes are selected to observe their output through the uncertainty analysis

Table 2 Object numbers of the SOTIF dataset

Category	Fog	Snow	Glare	Rain	Total
Pedestrian	28	39	94	166	327
Cyclist	12	5	55	32	104
Car	111	127	522	648	1408
Traffic signal	12	22	76	96	206
Total	163	193	747	942	2045

module. As shown on the left of Fig. 6, the weather in (a) is sunny with a little rain, the lighting condition in (b) is dark, and there is more water on the front windshield in (c). The bounding boxes predicted by the original model are extended by their standard deviations to draw the enveloping lines and the regions are lightly colored, as shown on the right of Fig. 6. As YOLO-416 regresses the central coordinates, the width, and the height of a bounding box, the drawn uncertainty region is centrosymmetric.

There is a missing prediction in Fig. 6a and a repetition prediction in Fig. 6b. The car on the left of Fig. 6a is almost blocked by the bushes, and the water droplet and the darkness confuse the model, making it generate two bounding boxes for the second car from the right of Fig. 6b. For the objects detected in both images, the uncertainty regions in Fig. 6b are generally larger than those in Fig. 6a, indicating that the dusky light condition increases the uncertainty of

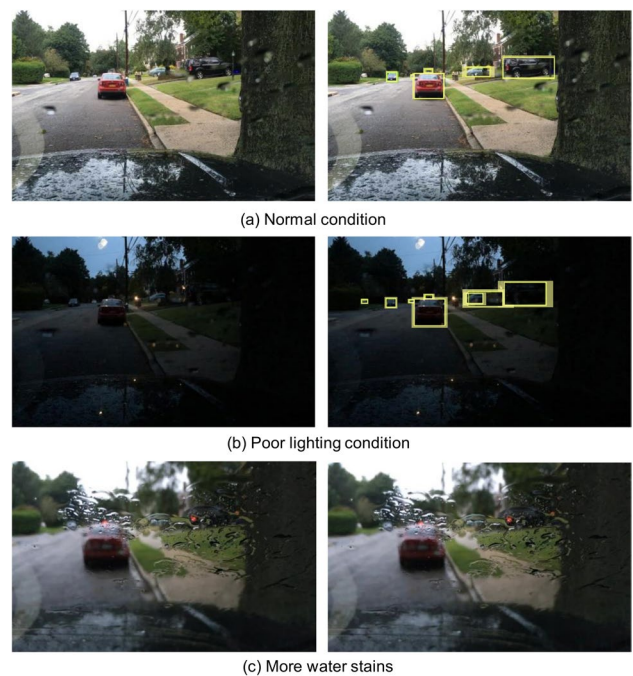


Fig. 6 Detection results of the case with visible uncertainty. Left: Raw images. Right: Object detector results with uncertainty region after clustering 20 samples

the detection results. As for Fig. 6c, no objects are successfully detected due to too much water on the front windshield, which blurs the characteristics of these cars. This situation can be alleviated by adding a rain removal algorithm for preprocessing [32]. Another feasible method is to identify this kind of danger using the perception fusion and transmit the command to start the windshield wiper.

In the test, it is also observed that the model with dropout layers is much slower than the original model in each run, not to mention several times of sampling. Therefore, the real-time performance of this method is far from that of real vehicles. Ideally, a dropout layer should be inserted after each hidden layer in the network as in the proposed method. However, it will significantly slow down the training and testing processes. Therefore, some researchers decided not to insert dropout layers into the shallow layers of the network to improve computing efficiency [23]. They believed that low-level features extracted from the shallow layers were mostly consistent and hence could be represented by deterministic parameters [24].

3.3 Performance and Uncertainty Evaluation

The YOLOv3 model with dropout layers added according to the MCD method is tested with the SOTIF dataset. Then, the test results are used in the evaluation system and the offline simulation results are obtained.

Firstly, the uncertainty information from the uncertainty analysis module is shown in Table 3, where the headers mean the abscissa and the ordinate of the center of the bounding box, the width and the height of the bounding box, the maximum probability of the categories, and the probability that there is exactly a certain object in the bounding box. For example, for a certain object in the Fog subset, if the model predicts a bounding box, then the abscissa of the center of the box has an average standard error of 0.0048 times the total width of the image.

Then, based on the results of uncertainty quantification, the confidence calibration is performed according to Fig. 3. The results are shown in Table 4. The data shows the effectiveness of confidence calibration. In Table 4, Acc is the actual accuracy obtained by comparing the output detection boxes and the ground truth. Con and CC are the average confidence levels of the model output before and after the calibration, respectively. Theoretically, the closer Con is to

Table 4 Results of confidence calibration

	Pedestrian	Cyclist	Car	Traffic signal
Acc_Fog	0.9693	0.9755	0.7607	0.9632
Con_Fog	0.8002	0.6602	0.7787	0.6705
CC_Fog	0.8490	0.6078	0.7744	0.7332
Acc_Snow	0.9741	0.9948	0.8394	0.9845
Con_Snow	0.7928	0.5630	0.7878	0.6390
CC_Snow	0.8293	0.6745	0.7916	0.7240
Acc_Glare	0.9424	0.9384	0.7162	0.9772
Con_Glare	0.7744	0.8126	0.7502	0.5678
CC_Glare	0.8094	0.7858	0.7438	0.6015
Acc_Rain	0.9204	0.9810	0.7197	0.9735
Con_Rain	0.7690	0.7293	0.7865	0.5653
CC_Rain	0.8080	0.7521	0.7719	0.6174

Acc, the more reliable Con is, i.e., the more credible the confidence of the model is [3]. Therefore, the calibrated CC is closer to Acc than Con whether Con is higher or lower than Acc, indicating that CC is more reliable as the model output. It can be observed from the table that in most instances, CC is much closer to Acc than Con. However, this is not the case in some other instances, as shown in Fig. 7. Specifically, the results of cyclists in fog and glare subsets are negative, which is probably due to the insufficient sample size and the ability of the model to distinguish cyclists from pedestrians.

The attempt of confidence calibration based on uncertainty is meaningful. Although more suitable parameters are not found to make the confidence level after calibration as close to the accuracy as possible, the changing trend of the confidence degree in most cases is correct. The results show that the model uncertainty may be related to the credibility of the output confidence, and the detailed correlation will be analyzed and determined.

Finally, the detection boxes of the model output are sorted by the calibrated confidence level, and the evaluation indexes such as mAP are further calculated. The results provide a reference for the improvement of the SOTIF of the algorithm. Since the validity of the confidence calibration process has been verified, the reliability of the evaluation results shown in Table 5 has also been further improved. The first few columns in Table 5 are the mAPs under different IoU thresholds, and the last column is their mean value. It can be observed that the performance of YOLO-416 applied

Table 3 Average standard deviation of the objects obtained by MCD ($\times 10^{-2}$)

Subset	X_{center}	Y_{center}	w	H	Probability	Objectness
Fog	0.48	0.95	1.42	2.70	9.00	7.94
Snow	0.65	0.71	2.19	2.31	9.75	8.72
Glare	0.49	0.76	1.43	2.14	9.85	9.06
Rain	0.46	0.99	1.55	2.55	9.85	8.54

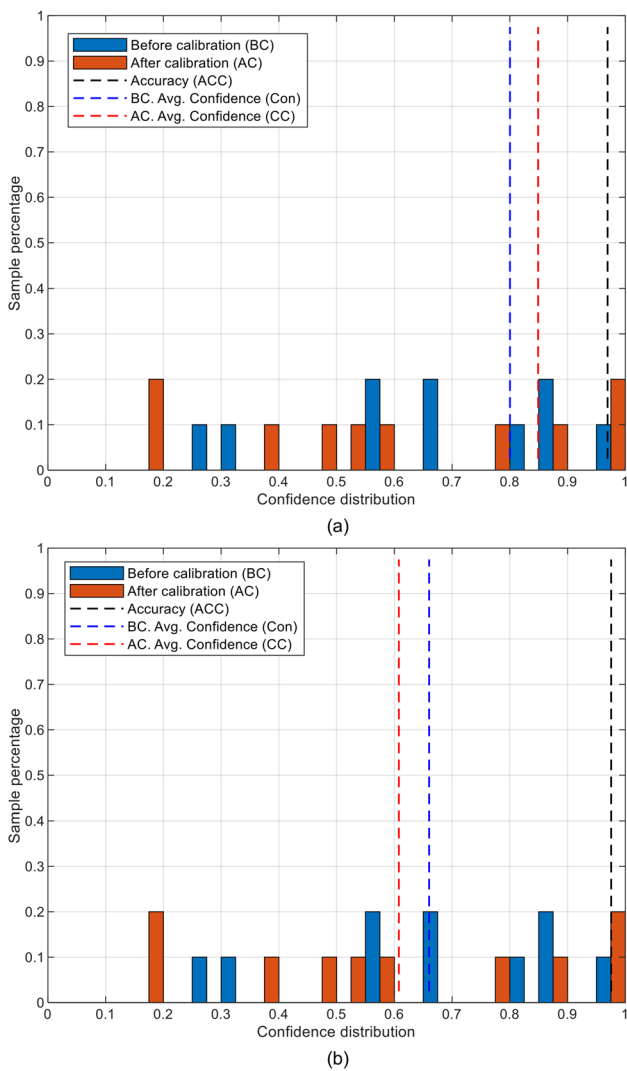


Fig. 7 Confidence distribution of **a** the pedestrian samples and **b** the cyclist samples in Fog subset before and after calibration

on the SOTIF dataset is not satisfactory, especially under the influence of glare. Therefore, additional measures are required such as using other algorithms in series to achieve performance breakthrough in these aspects before applying them on real vehicles.

Table 5 Calculation results of evaluation index mAP

IoU	0.5	0.6	0.7	0.8	0.9	mmAP
mAP_Fog	0.6175	0.5377	0.4283	0.2476	0.0271	0.3716
mAP_Snow	0.6860	0.6443	0.5122	0.2511	0.0177	0.4223
mAP_Glare	0.3428	0.3064	0.2313	0.0805	0.0098	0.1942
mAP_Rain	0.4286	0.3591	0.2482	0.0878	0.0126	0.2273

3.4 Tests on Extra Scenes

In the evaluation module, all the calibrated outputs of the entire dataset are input to obtain the mAP, which is the overall offline evaluation of the algorithm. If the method is applied to the real vehicles, it would be worth exploring which output information of the first three modules can be used to trigger the warning. There are many other scenes related to the perceptual SOTIF in the developed database, such as graffiti on traffic signs, stained lane marks, and potholes on the road surface. Four typical scenes are selected, as shown on the left of Fig. 8, where (a) and (c) refer to the abnormal postures of pedestrians, motorbikes, and cars, while (b) and (d) refer to the shape changes of vehicles and pedestrians caused by loads and carry-on objects, respectively. Then, the uncertainty regions of the objects of interest are drawn, as shown on the right of Fig. 8. Firstly, it can be observed that the model fails to detect the rollover truck and motorbike, and misjudges the truck loaded with bushes as a potted plant. This is the limitation of the algorithm that needs to be overcome by improving the algorithm or further training. Secondly, the uncertainty regions of the truck loaded with an elephant, the pedestrian on the ground, and the pedestrian holding an umbrella are significantly larger than the other normal objects. Therefore, an uncertainty threshold can be set, and once the uncertainty exceeds the threshold (e.g., 0.05), a warning will be triggered, indicating risks exist around the object. Lastly, another truck loaded with an elephant in Fig. 8b is also detected by the original model, and the uncertainty region is not large enough. However, it has only been detected three times in 20 sampling models. According to the design in Sect. 2.3, its calibrated confidence level is very low. Therefore, it is also necessary to set a confidence threshold. When the calibrated confidence is lower than the threshold (e.g., 0.5), a risk warning will be triggered, indicating that the model is not sure about the detection of the object.

3.5 Discussion

This section makes a summary and discussion about the methods and test results. Firstly, a preliminary database of SOTIF-related scenarios is established and some images with extreme weather or adverse lighting conditions are selected to form a SOTIF dataset. Secondly, the MCD

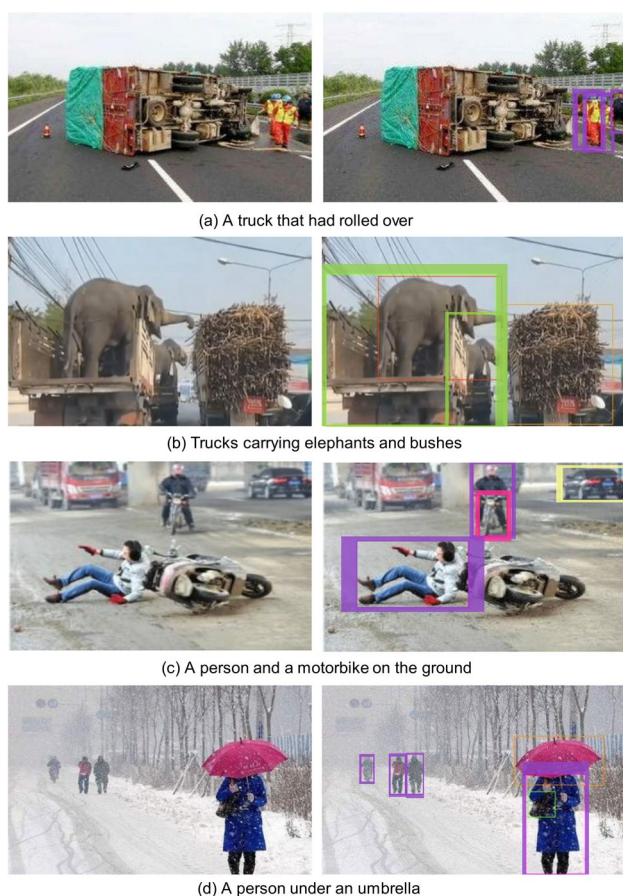


Fig. 8 Detection results of the scenes with visible uncertainty. **Left:** Raw images. **Right:** Object detector results with uncertainty region after clustering 20 samples

method is used to quantify the model uncertainty in the uncertainty analysis module. Then in the calibration module, some appropriate calibration factors are set to calibrate the confidence, and a method to transmit safety warnings to the decision-making and control systems is proposed. Lastly, the confusion matrix in the evaluation module is extended, considering both repetition and missing predictions in the object detection task. The results demonstrate and validate that the calibration module can make the average confidence of the model output closer to the actual accuracy in most cases. Therefore, the reliability of the evaluation results can be improved by using the calibrated confidence for mAP calculation.

It should be noted that there are differences between the applications of the MCD method on different works such as object detection and semantic segmentation. The position of each pixel in the semantic segmentation is fixed, while the object does not have an exact number to identify it. Therefore, it is difficult to match the bounding boxes of the

same object generated from different sampling models. To address this issue, a simple IoU comparator is applied, which is likely to fail when the objects are close to each other. If the bounding box of one object matches with the box of another object in that situation, the uncertainty estimation will be excessive. Moreover, when only the safety is considered and the accuracy of the uncertainty results is ignored, the potential risk caused by object aggregation can still be detected. However, when the object is at the boundary of danger and safety, the algorithm often considers it to be dangerous, and the subsequent decisions will be relatively conservative.

4 Conclusions

The Monte Carlo dropout method is applied to the deterministic YOLOv3 architecture to estimate the label and spatial uncertainty of the object detection algorithm. Experiments under SOTIF-related scenarios are carried out, and the results show that the uncertainties are usually high when the object itself is abnormal or affected by environmental factors such as rain and snow. Moreover, an optimization scheme of the mean Average Precision evaluation system for object detection algorithms is introduced. The improvements include using the uncertainties to the Histogram Binning method and expanding the confusion matrix according to the characteristics of the object detection task. The experiment results show that in most cases, the calibrated confidence is closer to the actual accuracy and the mAP evaluation index is therefore more reliable.

The authors identify five promising directions for future work that are currently being explored: (1) further analyze the relationship between the model uncertainty and the confidence level, and look for a more appropriate and effective confidence calibration method; (2) evaluate the estimated uncertainties themselves through some novel probabilistic metrics such as the Probability-based Detection Quality; (3) analyze and compare the model performance and uncertainty evaluation after introducing other uncertainty estimation methods to the YOLOv3 architecture (e.g., Deep Ensembles and Deep Evidential Regression); (4) analyze and compare the model performance and uncertainty evaluation after introducing the above methods to other object detection models (e.g., SSD and Retina Net); (5) continue to enrich the SOTIF-related scenario database by collecting videos and photos in which the objects are likely to be out-of-distribution and adversarial examples, and facilitate evaluating the SOTIF of various algorithms in autonomous driving.

Acknowledgements The authors would like to appreciate the financial support of the National Science Foundation of China Project:

U1964203 and 52072215 and National Key R&D Program of China: 2020YFB1600303.

Declaration

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- International Organization for Standardization: ISO/Pas 21448-road vehicles-safety of the intended functionality. Geneva, Switzerland (2019)
- Lin, C.H., Hsu, K.C., Johnson, K.R., et al.: Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry. *Comput. Meth. Programs Biomed.* **190**, (2020)
- Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q.: On calibration of modern neural networks. In: Paper presented at the 34th International Conference on Machine Learning, International Machine Learning Society, Sydney, 6–11 August (2017).
- Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: Paper presented at the 18th International Conference on Machine Learning, Williams College, Massachusetts, June 28 – July 1 (2001).
- Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Paper presented at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Edmonton, 23–26 July (2002).
- Naeini, M. P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using Bayesian binning. In: Paper presented at the 29th AAAI Conference on Artificial Intelligence, American Association for Artificial Intelligence, Austin, 25–30 January (2015).
- Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv Large Margin Classif* **10**(3), 61–74 (1999)
- Bhattacharyya, A., Fritz, M., Schiele, B.: Long-term on-board prediction of people in traffic scenes under uncertainty. In: Paper presented at the 31st IEEE Conference on Computer Vision and Pattern Recognition, Michael Brown, Salt Lake City, 18–22 June (2018).
- Michelmore, R., Wicker, M., Laurenti, L., Cardelli, L., Gal, Y., Kwiatkowska, M.: Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In: Paper presented at the 2020 IEEE International Conference on Robotics and Automation, Paris, France, May 31 – August 31 (2020).
- Neal, R.M.: MCMC using Hamiltonian dynamics. *Handbook Markov Chain Monte Carlo.* **2**(11), 2 (2011)
- Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: Paper presented at the 32nd International Conference on Machine Learning, Lille, France, 6–11 July (2015).
- Kahn, G., Villafior, A., Pong, V., Abbeel, P., Levine, S.: Uncertainty-aware reinforcement learning for collision avoidance. *Mach. Learn.* 1702.01182 (2017). <https://arxiv.org/abs/1702.01182v1>
- Zhu, L., Laptev, N.: Deep and confident prediction for time series at Uber. In: Paper Presented at the 17th IEEE International Conference on Data Mining Workshops, New Orleans, Los Angeles, 18–21 November (2017).
- Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: Paper Presented at the 33rd International Conference on Machine Learning, New York City, 19–24 June (2016).
- Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. In: Paper presented at the 31st IEEE Conference on Computer Vision and Pattern Recognition, Michael Brown, Salt Lake City, 18–22 June 2018.
- Wang, H., Khajepour, A., Cao, D., Liu, T.: Ethical decision making in autonomous vehicles: challenges and research progress. *IEEE Intell. Transp. Syst. Mag.* (2020). <https://doi.org/10.1109/MITS.2019.2953556>
- Wang, H., Huang, Y., Khajepour, A., Zhang, Y., Rasekhipour, Y., Cao, D.: Crash mitigation in motion planning for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **20**(9), 3313–3323 (2019)
- Kendall, A., & Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Paper presented at the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 21–26 July 2017.
- Osband, I.: Risk versus uncertainty in deep learning: bayes, bootstrap and the dangers of dropout. In: Paper presented at the 30th Conference and Workshop on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
- Lin, T. Y., Maire, M., Belongie, S., et al.: Microsoft coco: common objects in context. In: Paper presented at the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014.
- Hinton, G.E., Srivastava, N., Krizhevsky, A., et al.: Improving neural networks by preventing co-adaptation of feature detectors. *Comput. Sci.* **3**(4), 212–223 (2012)
- Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. *Adv. Neural. Inf. Process. Syst.* **29**, 1019–1027 (2016)
- Mukhoti, J., Gal, Y.: Evaluating Bayesian deep learning methods for semantic segmentation. In: Paper presented at the 32nd IEEE Conference on Computer Vision and Pattern Recognition Workshops 2019, Long Beach, California, 16–20 June 2019.
- Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segNet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In: Paper Presented at the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, 27–30 June 2016.
- Miller, D., Sünderhauf, N., Zhang, H., et al.: Benchmarking sampling-based probabilistic object detectors. In: Paper Presented at the 32nd IEEE Conference on Computer Vision and Pattern Recognition Workshops 2019, Long Beach, California, 16–20 June 2019.
- Miller D, Dayoub F, Milford M, et al.: Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In: Paper Presented at the 2019 IEEE International Conference on Robotics and Automation, Montreal, Quebec, 20–24 May 2019.
- Miller D, Nicholson L, Dayoub F, et al.: Dropout sampling for robust object detection in open-set conditions. In: Paper Presented

- at the 2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018.
28. Shao, L., Cai, Z., Liu, L., Lu, K.: Performance evaluation of deep feature learning for RGB-D image/video classification. *Inf. Sci.* **385**, 266–283 (2017)
 29. Geiger, A., Lenz, P. S., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. *Int. J. Rob. Res.* 32(11), 1–6 (2013).
 30. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: Bdd100k: a diverse driving video database for heterogeneous multitask learning. In: Paper Presented at the 33rd IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, 13–19 June 2020.
 31. Wishart, J., Como, S., Forgiione, U., Weast, J.: Literature review of verification and validation activities of automated driving systems. *SAE Int. J. Connect. Automat. Veh.* **3**(4), 267–323 (2020)
 32. Li, S., Ren, W., Zhang, J., Yu, J., Guo, X.: Single image rain removal via a deep decomposition-composition network. *Comput. Vis. Image. Underst.* **186**, 48–57 (2019)
 33. Wang, H., Huang, Y., Khajepour, A., Cao, D., Lv, C.: Ethical decision-making platform in autonomous vehicles with lexicographic optimization based model predictive controller. *IEEE Trans. Veh. Technol.* 69(8), 8164–8175 (2020).
 34. Azevedo, T., de Jong, R., Mattina, M., Maji, P.: Stochastic-YOLO: efficient probabilistic object detection under dataset shifts. In: Paper Presented at the 34th Conference and Workshop on Neural Information Processing Systems, Vancouver, Canada, 6–12 December 2020.