IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 January 2024, accepted 14 January 2024, date of publication 17 January 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355199

RESEARCH ARTICLE

Leveraging Monte Carlo Dropout for Uncertainty Quantification in Real-Time Object Detection of Autonomous Vehicles

RUI ZHAO[1], (Member, IEEE), KUI WANG[2], YANG XIAO[1], FEI GAO[1], AND ZHENHAI GAO[1], (Member, IEEE)

[1]College of Automotive Engineering, Jilin University, Changchun, Jilin 130025, China
[2]School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Yang Xiao (xiaoy2016@jlu.edu.cn)

This work was supported by the National Science Foundation of China under Grant 52202495 and Grant 52202494.

ABSTRACT With the recent advancements in machine learning technology, the accuracy of autonomous driving object detection models has significantly improved. However, due to the complexity and variability of real-world traffic scenarios, such as extreme weather conditions, unconventional lighting, and unknown traffic participants, there is inherent uncertainty in autonomous driving object detection models, which may affect the planning and control in autonomous driving. Thus, the rapid and accurate quantification of this uncertainty is crucial. It contributes to a better understanding of the intentions of autonomous vehicles and strengthens trust in autonomous driving technology. This research pioneers in quantifying uncertainty in the YOLOv5 object detection model, thereby improving the accuracy and speed of probabilistic object detection, and addressing the real-time operational constraints of current models in autonomous driving contexts. Specifically, a novel probabilistic object detection model named M-YOLOv5 is proposed, which employs the MC-drop method to capture discrepancies between detection results and the real world. These discrepancies are then converted into Gaussian parameters for class scores and predicted bounding box coordinates to quantify uncertainty. Moreover, due to the limitations of the Mean Average Precision (MAP) evaluation metric, we introduce a new measure, Probability-based Detection Quality (PDQ), which is incorporated as a component of the loss function. This metric simultaneously assesses the quality of label uncertainty and positional uncertainty. Experiments demonstrate that compared to the original YOLOv5 algorithm, the M-YOLOv5 algorithm shows a 74.7% improvement in PDQ. When compared with the most advanced probabilistic object detection models targeting the MS COCO dataset, M-YOLOv5 achieves a 14% increase in MAP, a 17% increase in PDQ, and a 65% improvement in FPS. Furthermore, against the state-of-the-art probabilistic object detection models for the BDD100K dataset, M-YOLOv5 exhibits a 31.67% enhancement in MAP and a 125.6% increase in FPS.

INDEX TERMS Uncertainty quantification, object detection, autonomous vehicles, YOLOv5, Monte Carlo dropout.

I. INTRODUCTION

In recent years, deep learning has been increasingly utilized in autonomous driving perception systems, where object detection models have made significant advancements in both result accuracy and inference speed [1], [2], [3]. However, in facing edge cases such as heavy snow, fog, rain, or extreme lighting conditions during the night, and unknown regular traffic participants, deep learning perception models are still likely to make incorrect predictions with a considerable probability [4], [5]. Fig. 1 illustrates the output results of the probabilistic object detection model in multiple traffic scenarios. The upper left portion represents a normal traffic scene, the upper right is under low-light conditions and

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

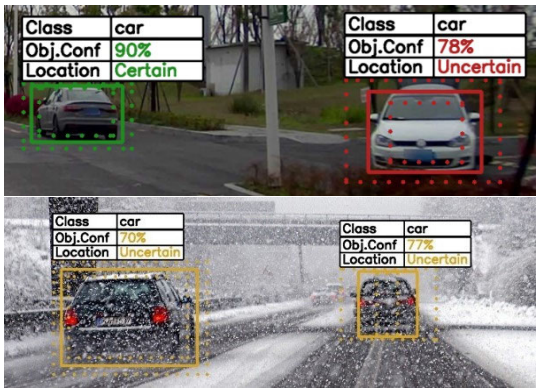


FIGURE 1. The figure compares traffic scenarios based on uncertainty. 'Obj.Conf' signifies object confidence, and 'Location' the uncertainty in predicted bounding box location. Prediction box colors indicate certainty levels: green for certainty, yellow for medium confidence, and red for extreme uncertainty. Solid lines show the bounding box mean, while dashed lines represent a 90% confidence range.

the lower half depicts extreme weather conditions, during which the location of the object detection model's output is largely uncertain. Corresponding safety redundancy in cognition and decision-making must be implemented based on the quantified uncertainty. Acquiring the uncertainty in perception model predictions can provide valuable information to the decision-making layer and assist autonomous vehicles in taking timely actions. Furthermore, human beings have an intuitive ability to understand uncertainty, so perceptual uncertainty information can help better interpret the intentions of autonomous vehicles and increase trust in autonomous driving technology [6]. Accurately quantifying the uncertainty of perception model detections has become a necessary condition for solving the safety long-tail effect in autonomous driving.

The current mainstream autonomous driving object detection models primarily include the SSD series, R-CNN series, YOLO series, and models based on Transformer network structures [7], [8], [9], [10], [11], [12]. Early SSD series models were fast but had low detection accuracy, failing to meet the requirements of autonomous driving perception systems. The emergence of R-CNN series detectors, such as Fast R-CNN [7], Faster R-CNN [8], RFCN [9], compensated for the low detection accuracy but significantly increased detection time, not meeting the real-time requirements of autonomous driving perception systems.

In pursuit of balancing accuracy and speed in object detection, numerous algorithms have evolved, notably the YOLO series [10]. These models segment images into grids for simultaneous multi-object detection and have undergone five iterations to date. YOLOv1 was limited in localizing small or overlapping objects. YOLOv2, utilizing Darknet-19 and anchor boxes, enhanced localization but still struggled with small objects. YOLOv3, with Darknet-53 and varied feature map sizes, improved small object detection at a reduced speed. YOLOv4 integrated technologies like Complete Intersection over Union (CIoU), achieving higher accuracy without compromising the speed of YOLOv3. YOLOv5, incorporating the SPPF module, minimized hardware needs, gaining widespread application in academia and industry. Following YOLOv5, YOLOv6, YOLOv7, and YOLOv8 were introduced. YOLOv6, by Meituan's Vision AI, featured an efficient design with advanced components. YOLOv7, from YOLOv4 and YOLOR authors, excelled in speed and accuracy, demanding high computational resources. YOLOv8 utilized an anchor-free, decoupled head design for improved accuracy (AP 53.9%) but required significant computational power and training time. After YOLOv5, the series achieved higher accuracy but at the cost of increased computational demands and limited industrial applicability.

Currently, YOLOv5 remains prevalent in autonomous driving perception systems due to its rapid detection speed and high accuracy. In the current technological framework of major automotive companies, such as Tesla, Audi, BMW, and Mercedes-Benz, YOLOv5 is employed for comprehensive object detection, vehicle detection, pedestrian detection, and lane line detection within the realm of autonomous driving. This utilization positions YOLOv5 as a pivotal algorithm in the perception systems of autonomous vehicles [11], [12], [13], [14]. YOLOv5's robustness, proven through its long-term validation and critical use in the automotive industry, extends beyond autonomous driving to other sectors. Its applications range from detecting defects in manufacturing, managing traffic and parking in transportation, to identifying irregularities in medical imaging [15], [16], [17].

Subsequently, the Transformer has profoundly impacted the entire field of deep learning, particularly in computer vision. To overcome the limitations of CNNs, transformer algorithms have abandoned traditional convolutional operators, instead opting for attention mechanisms alone, achieving a global scale receptive field. Recently, Chen et al. [18] proposed a hybrid network transformer based on the Transformer for object detection, achieving superior accuracy. Concurrently, Qi et al. [19] introduced an integration of multi-scale feature extraction with transformers model in single-stage object detection, effectively balancing detection speed and precision. Additionally, Yuan et al. developed a transformer-based object detection algorithm tailored for autonomous driving [20]. Although these algorithms demonstrate commendable detection capabilities, their limitations in uncertainty modeling render them less suitable for application in safety-critical domains such as autonomous driving. For instance, they exhibit overconfidence in detection results when faced with dynamic traffic participants with high randomness.

Object detectors, which are primarily accuracy-focused, struggle in edge cases like severe weather, risking incorrect autonomous driving decisions [21], [22]. Their key limitation is poor uncertainty judgment in obstacle identification. This overconfidence in predictions can lead to safety issues. Research now emphasizes enhancing machine learning models in safety-critical areas, like autonomous driving, by improving uncertainty estimation. In autonomous driving, probabilistic object detection models assess output

uncertainty, utilizing methods like error propagation, direct modeling, ensemble methods, and MC-drop methods [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37].

Error propagation [23] calculates biases cumulatively but isn't ideal for autonomous systems due to its emphasis on model confidence over actual uncertainty. Direct modeling expands detector output dimensions and is integrated with detectors like SSD [24] and RCNN [25], [26], [27], offering high-quality uncertainty estimates and fast detection but increasing training resource demands. Ensemble methods, applied by researchers like Guo and Gould [28], Lin et al. [29] and Lee et al. [30], train multiple networks and compare outputs to estimate uncertainty. These methods match direct modeling in estimation quality but require significant computational resources.

The MC-drop method [31], [32] leverages pretrained weights for Gaussian modeling without needing network changes. Initially applied to SSD by Miller et al. [33], it was later adapted for [34] for better accuracy, though at slower detection speeds. Choi et al. [35] effectively merged MC-drop with YOLOv3, balancing speed and accuracy. Azevedo et al. [36] further refined this, enhancing speed and uncertainty estimation. The Bayes OD model [37] then emerged, targeting the BDD100K dataset with Bayesian methods for improved detection accuracy and uncertainty assessment, but at the cost of computational efficiency, challenging its use in real-time autonomous driving. Despite progress in probabilistic object detection, these methods still trail behind accuracy-centric models in autonomous systems, underscoring the need for uncertainty quantification.

Uncertainty quantification in object detection is a recent development lacking a unified standard. Traditional evaluation, like MAP [38], ranks predicted boxes by confidence scores and measures localization accuracy via IoU. AP is derived from the P-R curve, and MAP is the mean AP across categories. The mean MAP over different IoU thresholds indicates overall accuracy. However, MAP has limitations in evaluating perception model confidence, often misrepresenting the correlation between confidence and accuracy. This leads to overconfidence or under confidence in predictions, making it unsuitable for probabilistic object detectors. Recently, Hall et al. [39] introduced PDQ, a new metric for evaluating label and spatial uncertainty in object detection, which appears more appropriate for evaluating probabilistic models in autonomous driving, but is yet to be widely adopted in related research.

The current research issues are summarized as follows: Firstly, the main development direction of object detection algorithms has been towards improved accuracy and speed. However, in safety-critical applications such as autonomous driving, there is an urgent need for probabilistic object detection algorithms that concurrently offer real-time performance, accuracy, and uncertainty estimation – an area where current research is relatively lacking. Secondly, research on uncertainty assessment in object detection is still in its exploratory phase. The MC-drop method shows potential for high fidelity and accuracy in uncertainty estimation. Yet, current research does not sufficiently explore how internal parameters of the MC-drop method affect model performance. This, combined with unidimensional performance evaluations, leads to suboptimal uncertainty assessment results when using the MC-drop method.

To address gaps in current research, we introduce M-YOLOv5, a probabilistic object detection model that offers enhanced detection speed, accuracy, and uncertainty assessment over existing models. Despite YOLOv5's industrial popularity, its lack of interpretability limits its autonomous driving application. M-YOLOv5 integrates YOLOv5 with uncertainty modeling, employing an adapted MC-drop method to estimate class and location uncertainties using Gaussian parameters. This adaptation improves its applicability in autonomous driving. We also propose the PDQ metric, a more effective alternative to the traditional MAP system. Our study includes a sensitivity analysis of crucial hyperparameters like Dropout probability and layer configuration, impacting uncertainty estimation and accuracy. Extensive experiments validate M-YOLOv5's superiority in probabilistic object detection.

The main contributions and innovations of this paper are as follows:

- This research introduces M-YOLOv5, a sophisticated probabilistic object detection algorithm utilizing the MC-drop method to adeptly identify anomalies in image data, inaccuracies in neural network-extracted image features, and variances between network perceptions and actual detections, effectively quantifying the inherent uncertainty in detection results. It also resolves the stringent real-time performance requirements in autonomous driving through a cleverly designed structure. This algorithm can be directly applied in the field of autonomous driving and possesses strong portability.
- A sensitivity analysis of significant hyperparameters in the MC-drop method has been conducted to identify the optimal way of incorporating Dropout layers into object detection models. To our knowledge, there has not yet been any research exploring the sensitivity analysis of the positions and quantities of Dropout layers in object detection models.
- Extensive experiments were conducted, showing that compared to the original YOLOv5 algorithm, the M-YOLOv5 algorithm achieves a 74.7% improvement in PDQ. Against Hall et al.'s probFRCNN [39], a state-of-the-art probabilistic object detection model targeting the MS COCO dataset, M-YOLOv5 demonstrates a 14% increase in MAP, a 17% increase in PDQ, and a 65% increase in Frames Per Second (FPS). Furthermore, compared to the advanced probabilistic object detection model proposed by Feng et al. [37] for the BDD100K dataset, M- YOLOv5 shows a 31.67% improvement in MAP and a 125.6% increase in FPS.

The remainder of this paper is organized as follows: Section II outlines the design of the M-YOLOv5 model, including problem definition, constructed model, and model evaluation indicators. Section III conducts a sensitivity analysis of the hyperparameters that greatly influence the performance of MC-drop methods. Section IV presents extensive experiments and discusses the results, highlighting the superiority of the M-YOLOv5 algorithm. Finally, conclusions and future work are provided in Section V.

II. M-YOLOv5 MODEL

This section describes the proposed probabilistic object detection algorithm M-YOLOv5, which employs the MC-Drop method to incorporate class uncertainty and bounding box location uncertainty into the model's predictions. The section begins by defining the problem, followed by an introduction to the network structure of M-YOLOv5, which includes the CSPnet structure, the design of the MC-Drop method, and the process of uncertainty quantification. Subsequently, the design of the loss function is elaborated, and finally, the computation of the PDQ evaluation metric is detailed.

A. PROBLEM FORMULATION

This work aims to perform uncertainty modeling on the YOLOv5 model. Assuming that there are existing input data for object detection, the YOLOv5 model, and the original YOLOv5 network weights that have been trained, the task is to quantify the uncertainty in label and location of the YOLOv5 detection results.

To appropriately define this problem, specific symbols and parameters are first introduced. Let a labeled test set comprising N pairs of data be represented as $T = \{\boldsymbol{d}_n, \boldsymbol{r}_n\}_{n=1}^{N}$, where $\boldsymbol{d}_n$ is randomly selected input image data from the set D, and $r_n = \{\bar{r_c}, \bar{r_l}, r_p\}$ corresponds to the target output data from the object detection result set R. Here, $\bar{r_c}$ represents the type of object and the probability of each class, $\bar{r_l}$ represents the position of the object in the image, and r_p denotes the uncertainty of the detection result. Let $c\epsilon\{1, 2, \ldots, C\}$ represent the category code corresponding to the target, where C is the total number of target classes. Let $i\epsilon\{1, 2, \ldots, I\}$ indicates the current number of samples, where I represents the sampling times of the object detector. Let $obj_i\epsilon\{1, \ldots, C\}$ represents the class of the object, and $p_{0i}, p_{1i}, \ldots, p_{Ci}$ respectively represent the probability of each class. Then define $\bar{r_c} = \frac{\sum_{i=1}^{I} r_c^i}{I}$, where $r_c^i = \left(obj_i, p_{0i}, p_{1i}, \ldots, p_{Ci}\right)$. Let x_i, y_i represent the coordinates of the center of the predicted box, w_i the width, and h_i the height of the box, then $\bar{r_l}$ is expressed as $\bar{r_l} = \frac{\sum_{i=1}^{I} r_l^i}{I}$, where $r_l^i = (x_i, y_i, w_i, h_i)$. Define $p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right)$ as the probability that the input data $\boldsymbol{d}$ leads to an object class of $\bar{r_c}$ under a specific object detection model, and $p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)$ as the probability that the object location is $\bar{r_l}$, then r_p^i is expressed as $r_p^i = \left(p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right), p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)\right)$.

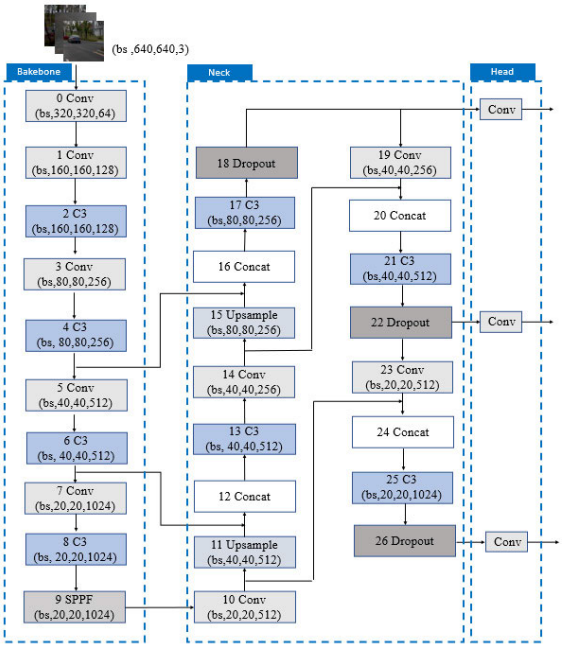


FIGURE 2. M-YOLOv5 model structure diagram.

The goal of this paper is to provide an accurate estimate of object detection class uncertainty $p_c\left(r_c^i|\boldsymbol{d}, \mathcal{D}\right)$ and location uncertainty $p_l\left(r_l^i|\boldsymbol{d}, \mathcal{D}\right)$, along with the detection class results $\bar{r_c}$ and location results $\bar{r_l}$, based on the original object detection model f, by designing the MC-drop method, and according to the input image data $\{\boldsymbol{d}_n\}_{n=1}^{N}$.

B. NETWORK STRUCTURE

To ensure the unambiguous safety compliance of ego vehicle, The network structure of M-YOLOv5 consists of three parts: Backbone, Neck, and Head, as illustrated in Fig. 2. The Backbone structure is responsible for extracting key features from the image, the Neck is tasked with fusing the extracted image features, and the Head part is in charge of transforming the fused features into the data's output format. To ensure that the features extracted by the Backbone structure are not disrupted, a Dropout layer is embedded between the Neck and Head structures.

When an image is input into the M-YOLOv5 network as the first layer input d^0 of the CNN, it first passes through the Backbone layer, resulting in the input d_{neck} to the Neck structure:

$$d_{neck} = Backbone\,(d) \tag{1}$$

The Backbone network structure is crucial for extracting image features, and its output is a linear or nonlinear combination of the intermediate layer outputs. Therefore, the output of a k-layer CNN can be represented as:

$$d_{neck} = F\left(d^0\right) = H_k\left(d^{k-1}, H_{k-1}\left(d^{k-2}\right), \ldots, H_1\left(d^0\right)\right) \tag{2}$$

where F represents the CNN network model, and H_k is the operation function of the k-th layer in the network structure.

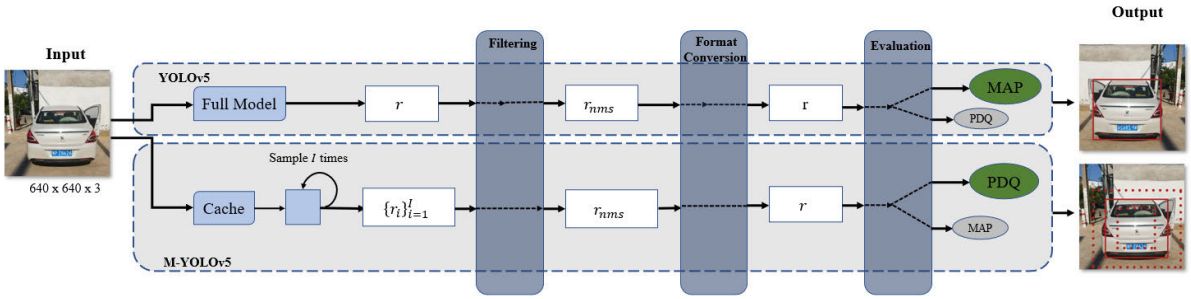


FIGURE 3. Description of the key building blocks of the YOLOv5 and M-YOLOv5 models, including prediction, format conversion, and evaluation. M-YOLOv5 first obtains $\{r_i\}_{i=1}^{I}$ through multiple sampling, then acquires r_{nms} via non-maximum suppression, and finally attains r through format conversion. Output detections for 2D images are visualized as bounding box mean (line) and bounding box extent at 90% confidence (dashed line).

To avoid gradient accumulation leading to the relearning of redundant information, the Backbone network structure focuses optimization on each layer's network model H_i, and the output of the k-th layer is expressed as:

$$d^k = M\left(\left[d^{k-1'}, T\left(F\left(d^{k-1''}\right)\right)\right]\right) \tag{3}$$

Here, $d_{k-1'}$ and $d_{k-1''}$ are two parts of d_{k-1} divided along the channel, T is a transition function truncating the gradient flow of $H_1, H_2, \ldots, H_k$, and M is a transition function used to blend the two segmented parts. The Backbone block comprises five convolutional layers *Conv*, four connecting layers $C3$, and a fast Spatial Pyramid Pooling Fast layer *SPPF*. SPPF is a pooling strategy that transforms feature maps of varying sizes into vectors of a fixed length. This is achieved by performing pooling operations at multiple scales and concatenating the results into a single feature vector. Additionally, this structure has been optimized to enhance the operational speed of the model. The output of the Backbone network structure will serve as the input for the Neck network structure.

The primary function of the Neck network structure is to fuse and optimize the features obtained from the Backbone at multiple scales, thus providing richer and more discriminative features for subsequent object detection. Specifically, the Neck network structure addresses the issue of scale invariance in object detection. By embedding a Dropout layer after the Neck structure, it ensures that the Dropout layer does not disrupt the image features extracted by the Backbone. When the Neck network structure receives the input from the Backbone network, it leads to the Dropout module's input $d_{dropout}$:

$$d_{dropout} = Neck\left(d_{neck}\right) \tag{4}$$

The Neck block, aside from the Dropout layer, includes convolutional layers *Conv*, connecting layers $C3$, fusion layers *Concat*, and upsampling layers *Upsample*. It comprises three output results that are fed into the "Head" block, corresponding to the detection of large, medium, and small objects in the final target detection outcome. Notably, the first column of fusion layers in the Neck block is integrated from different positions of the Backbone block, enabling a more comprehensive and effective capture of the image's features.

The MC-drop method can approximate the posterior distribution of Bayesian inference through the Dropout method, and thereby quantify the uncertainty of the object detection model using Bayesian inference. Upon inputting $d_{dropout}$ into the Dropout layer, the input d_{head} to the detection head is obtained:

$$d_{head} = d_{dropout} * diag\left[m_j^i\right]_{j=1}^{K_i} \tag{5}$$

$$m_j^i \sim Bernoulli\,(p) \quad for\ j = 1, \ldots, K_i \tag{6}$$

where m_j^i represents the j-th neuron in the i-th layer, with a value of 0 indicating that the neuron is in an inactive state, and a value of 1 indicating that it is normal. It follows a Bernoulli distribution with a probability of p. After processing through the Dropout layer, a new model weight matrix will be obtained, and for a given input, there will be different output results. Subsequently, d_{head} is input into the Head to obtain the raw output results r_i without non-maximum suppression:

$$r_i = Head\left(d_{head}\right) \tag{7}$$

Here, i represents the number of sampling times.

Fig. 3 illustrates the uncertainty quantification process of M-YOLOv5. To accelerate the uncertainty quantification speed of the M-YOLOv5 model, the network structure preceding the Dropout layer is run once for each detection, and the results are cached. Then, the cached results are sampled I times through the Dropout layer and the subsequent network. Since the network parameters before the Dropout layer are determined, caching to reduce the running times of the network structure before the Dropout layer will not affect uncertainty prediction. After the sampling is completed using the M-YOLOv5 algorithm, $\{r_i\}_{i=1}^{I}$ is obtained. This is followed by non-maximum suppression to yield r_{nms}, and then a format conversion is performed to produce the output result r, which is suitable for PDQ evaluation.

Because the M-YOLOv5 model's results will have a large number of overlapping bounding boxes in each sampling, NMS technology is needed to obtain the highest-scoring prediction box. The functions of NMS include: removing prediction boxes with confidence below a certain threshold

α; removing prediction boxes with at least one coordinate outside the image boundary; retaining the bounding box with the higher classification score and removing the other one when two prediction boxes have an IOU intersection greater than u.

The output format of the M-YOLOv5 model after each sampling, but before non-maximum suppression, is as follows:

$$r_i = \{x_i, y_i, w_i, h_i, obj_i, p_{0.i}, \ldots, p_{C.i}\} \tag{8}$$

where i represents the number of samples, and then the result is transformed into:

$$mean = \left\{\bar{x}, \bar{y}, \bar{w}, \bar{h}, \overline{p_0}, \ldots, \overline{p_C}\right\} \tag{9}$$

Through n sampling iterations, the covariance matrix $\sum i$ of the bounding box's upper-left and bottom-right coordinate values is calculated. After r_i undergoes non-maximum suppression, the first four items are taken, namely the center coordinates of the prediction box as well as its width and height. These values are then transformed into the coordinates of the two diagonal points of the prediction box.

$$b_i = \{x_{1.i}, y_{1.i}, x_{2.i}, y_{2.i}\} \tag{10}$$

Then, the covariance matrices for the two coordinate values are computed:

$$C_t = \begin{bmatrix} \Sigma_i xx & \Sigma_i xy \\ \Sigma_i yx & \Sigma_i yy \end{bmatrix} \tag{11}$$

where $t = 1, 2$ represent the upper-left and bottom-right coordinates, respectively. Since b_i represents the two diagonal coordinates of the predicted box, the two covariance matrices C_1 and C_2 must be calculated. These will be used in the subsequent calculation of the PDQ process. If the covariance matrix is not positive semi-definite, we transform it by calculating the eigenvalue decomposition and then reconstruct the matrix where previously negative eigenvalues are set to zero. The mean is then transformed after non-maximum suppression as follows:

$$r_{nms} = \{x, y, w, h, obj, p_0, \ldots, p_C\} \tag{12}$$

Combine it with the covariance matrix computed from Equation (11), and calculate the label uncertainty $p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right)$ and the location uncertainty $p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)$ using the Gaussian distribution through C_1, C_2 and the scores of each category. Finally, transform the results into.

$$r = \{x, y, w, h, obj, p_0, \ldots, p_c, p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right), p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)\} \tag{13}$$

where x, y, w, h represent the final predicted box's center coordinates and its width and height. The object category corresponding to this predicted box is obtained by the model through $max\{p_0, \ldots, p_c\}$.

Algorithm 1 M-YOLOv5 and Calculate PDQ

Input: $PictureData = \boldsymbol{d}$
Output: PDQ score and detection result r
$i = 1$;
$Cacsh \leftarrow Picture$;
for $i \leq 10$ **do**
$d_{neck} = Backbone\,(d)$;
$d_{neck} = F\left(d^0\right) = H_k\left(d^{k-1}, H_{k-1}\left(d^{k-2}\right), \ldots, H_1\left(d^0\right)\right)$
$d^k = M\left(\left[d^{k-1'}, T\left(F\left(d^{k-1''}\right)\right)\right]\right)$;
$d_{dropout} = Neck\,(d_{neck})$
$d_{head} = d_{dropout} * diag\left[m_j^i\right]_{j=1}^{K_i}$
$m_j^i \sim Bernoulli\,(p)\,forj = 1, \ldots, K_i$;
$r_i = Head\,(d_{head})$
$r_i = \left\{x_i, y_i, w_i, h_i, obj_i, p_{0.i}, \ldots, p_{c.i}\right\}$;
$mean = \sum_{i=0}^{10} r_i/10, i++$;
$b_i = \{x_{1.i}, y_{1.i}, x_{2.i}, y_{2.i}\}$;
$C_t = \begin{bmatrix} \sum_i xx & \sum_i xy \\ \sum_i yx & \sum_i yy \end{bmatrix}$;
Calculate $p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right), p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)$ and r_{nms} based on C_1, C_2;
$r = r_{nms} + p_c\left(\bar{r_c}|\boldsymbol{d}, \mathcal{D}\right) + p_l\left(\bar{r_l}|\boldsymbol{d}, \mathcal{D}\right)$;
Calculate D_i^n according to r and G_i^n is ground-truths;
for $pairs(D_i^n, G_i^n)$ **do**
calculate $Q_s(D_i^n, G_i^n)$;
calculate $Q_L(D_i^n, G_i^n)$;
$pPDQ\left(D_i^n, G_i^n\right) = \sqrt{Q_s \times Q_L}$;
Calculate PDQ;
return (PDQ, r);
end for

C. LOSS FUNCTION

For the problem of object detection, a good bounding box loss function should include three factors: overlap area, center distance, and aspect ratio. M-YOLOv5 adopts the CIoU loss function, which takes these three factors into account simultaneously, and its penalty term can be expressed as:

$$\mathcal{R}_{CIou} = \frac{\rho^2\left(\boldsymbol{b}, \boldsymbol{b}^{gt}\right)}{c^2} + \alpha v \tag{14}$$

where α is a positive balancing parameter, $\boldsymbol{b} = \{x, y\}$ and $\boldsymbol{b}^{gt} = \left\{x^{gt}, y^{gt}\right\}$ represent the center coordinates of the predicted bounding box B and the ground truth bounding box B^{gt} t, respectively. $\rho(\cdot)$ denotes the Euclidean distance, c is the diagonal length of the smallest bounding box that encompasses both boxes, and v measures the consistency of the aspect ratio of the detected bounding box.

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{15}$$

where w, h, represent the width and height of the predicted bounding box, respectively, and w^{gt}, h^{gt} represent the width and height of the ground truth bounding box. Consequently, the CIoU is defined as:

$$\mathcal{L}_{CIoU} = 1 - IoU + \mathcal{R}_{CIou} \tag{16}$$

where IoU is defined as:

$$IoU = \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|} \tag{17}$$

A positive tradeoff α is defined as:

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{18}$$

D. PROBABILITY-BASED DETECTION QUALITY

As a measure for assessing the quality of two-dimensional probabilistic object detection in images, PDQ aims to jointly evaluate label uncertainty and spatial uncertainty in image-based object detection. The evaluation of label uncertainty is achieved by matching the predicted classification scores with the ground-truth labels for each object instance in the image. Spatial uncertainty is calculated through covariance matrices, assuming a Gaussian distribution for the top-right or bottom-left corners of the bounding box. Optimal PDQ can be obtained when a prediction probability is correlated with prediction error, for instance, when a larger spatial uncertainty is associated with an inaccurate bounding box prediction. PDQ utilizes the Hungarian algorithm to assign an optimal corresponding detection for each ground-truth value, eliminating the dependency on the IOU threshold required by MAP. Additionally, PDQ measures the probability quality assigned to true positive detection results and evaluates it on a single classification score threshold, requiring object detection algorithms to filter low-scoring output detection results before assessment. The specific calculation method for PDQ is as follows:

$$PDQ\left(G, D\right) = \frac{1}{\sum_{n=1}^{N_F} N_{TP}^n + N_{FN}^n + N_{FP}^n} \sum_{f=1}^{N_F} \sum_{i=1}^{N_{TP}^n} q^n\left(i\right) \tag{19}$$

where N_{TP}^n, N_{FN}^n.N_{FP}^n represent the numbers of true positives, false negatives, and false positives detected by the detector at that frame number, respectively. $q^n = \left[q_1^n, \ldots, q_{N_{TP}^n}^n\right]$ represents the collection of non-zero pPDQ values at that frame number.

The value of pPDQ is determined by two parts: label quality and spatial quality, and the calculation formula is as follows:

$$\text{pPDQ}\left(G_i^n, D_i^n\right) = \sqrt{Q_s\left(G_i^n, D_i^n\right) \cdot Q_L\left(G_i^n, D_i^n\right)} \tag{20}$$

where G_i^n represents the set of the i-th ground truth objects in frame f, this set includes the actual bounding box, class label, and the object's segmentation mask itself.

D_i^n is the set of the i-th detected objects in frame f, which includes a probability function, the detected segmentation mask (with non-zero pixels), and scores for all possible class labels. Q_s denotes the spatial quality:

$$Q_s\left(G_i^n, D_i^n\right) = \exp\left(-\left(L_{FG}\left(G_i^n, D_i^n\right) + L_{BG}\left(G_i^n, D_i^n\right)\right)\right) \tag{21}$$

where L_{FG} is the foreground loss, representing the detector's average negative log-probability assigned to the pixels of the ground truth object. L_{BG} is the background loss, penalizing any probability mass that the detector erroneously assigns to pixels outside the ground truth bounding box. Q_s takes the maximum value of 1 when the detector allocates a probability of 1 to all true pixels within the ground truth. These two components can be calculated through matrix association. For details of the computation, readers are referred to [39].

Spatial quality describes the goodness of an object's location within the image, while label quality Q_L describes the effectiveness of the detection in recognizing what the object is. Q_L is the detector's probability estimate for the ground truth class of the object, regardless of whether this class ranks highest in the detector's probability distribution.

It is calculated as follows:

$$Q_L\left(G_i^n, D_i^n\right) = I_j^n\left(\hat{c}_i^n\right) \tag{22}$$

Unlike MAP this value is explicitly used to influence the quality of detection, rather than merely ranking the detector's predicted label probabilities without considering the actual label probabilities. The PDQ score can evaluate the detector'soverall performance in terms of both label uncertainty and spatial uncertainty.

The running process of the M-YOLOv5 model and the algorithmic procedure for PDQ computation are shown in Algorithm.1. Within the algorithm, the Model, representing the M-YOLOv5 model, is divided into two parts: cache and last. ''Cache'' refers to the network structure prior to the first Dropout layer, and ''last'' refers to the first Dropout layer and all subsequent network structures. The algorithm takes as input the image data that needs to be detected and outputs the detected categories, location, quantified uncertainty, and PDQ score. Lines 1 to 14 describe the prediction and uncertainty quantification process of the M-YOLOv5 model, while lines 15 to 19 detail the calculation process for the PDQ.

III. SENSITIVITY ANALYSIS OF MC-DROP METHOD

The design key to the MC-drop uncertainty modeling method lies in the placement of the Dropout layer, the number of Dropout layers, and the Dropout probability. Therefore, we conduct a sensitivity analysis on these key influencing factors. We first carry out a sensitivity analysis for the location of the Dropout layer and the Dropout probability. To avoid disrupting the effective sampling process of the YOLOv5 model, we only position it after different modules at the detection head. The experiment analyzed the effect of the Dropout layer's position on MAP, PDQ, *Avg_label*, and *Avg_spatial*, where *Avg_label* represents the average label quality and *Avg_spatial* represents the average spatial quality. Fig. 4 shows the sensitivity analysis results for Dropout probability and Dropout layer location. Each plot contains three curves corresponding to Dropout probabilities $p = 0.15$, $p = 0.2$, $p = 0.25$; the horizontal axis represents the position where the Dropout layer is added, and the vertical axes

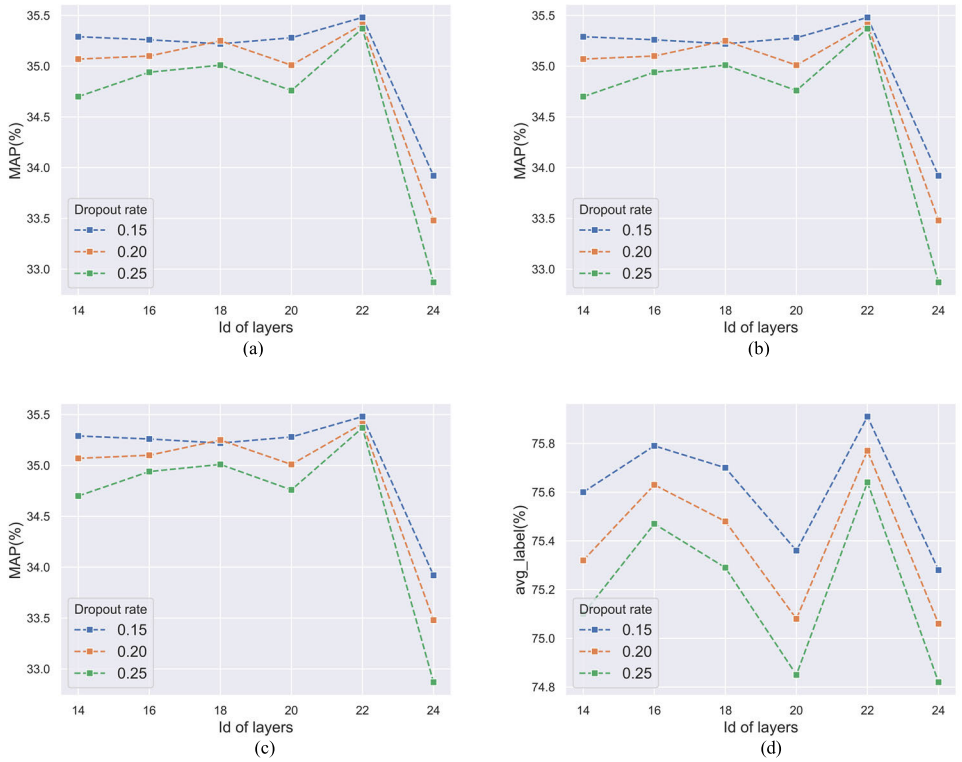


FIGURE 4. Dropout probability and dropout location sensitivity analysis, a MAP, b PDQ, c Avg_label and d Avg_spatial at MS COCO.

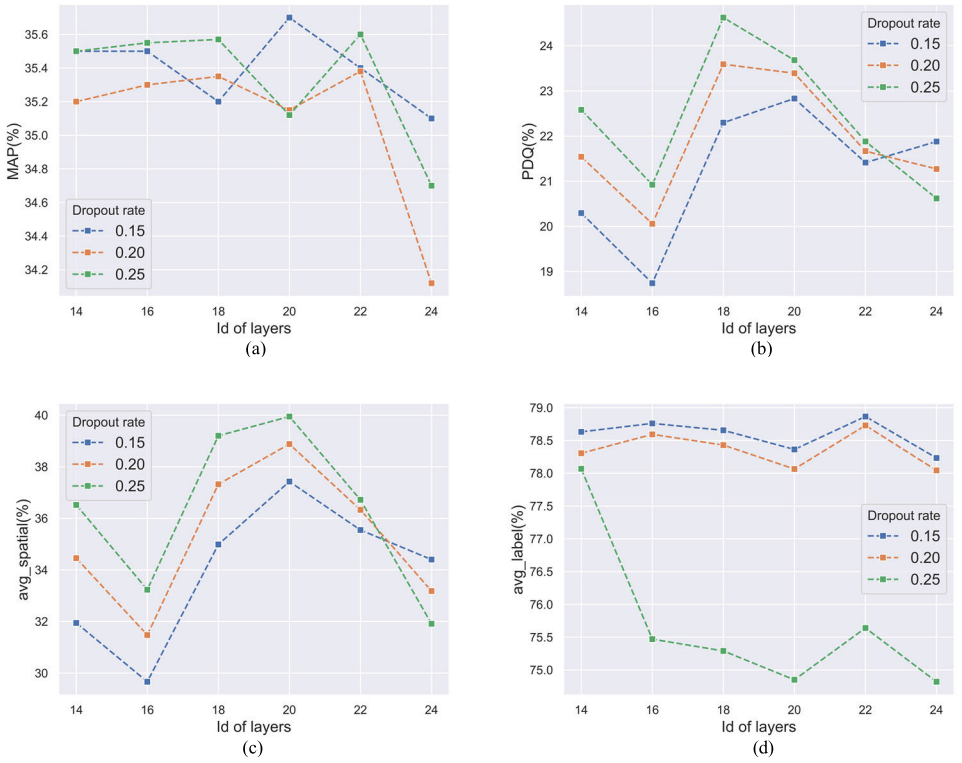


FIGURE 5. Dropout probability and dropout location sensitivity analysis, a MAP, b PDQ, c Avg_label and d Avg_spatial at MS COCO.

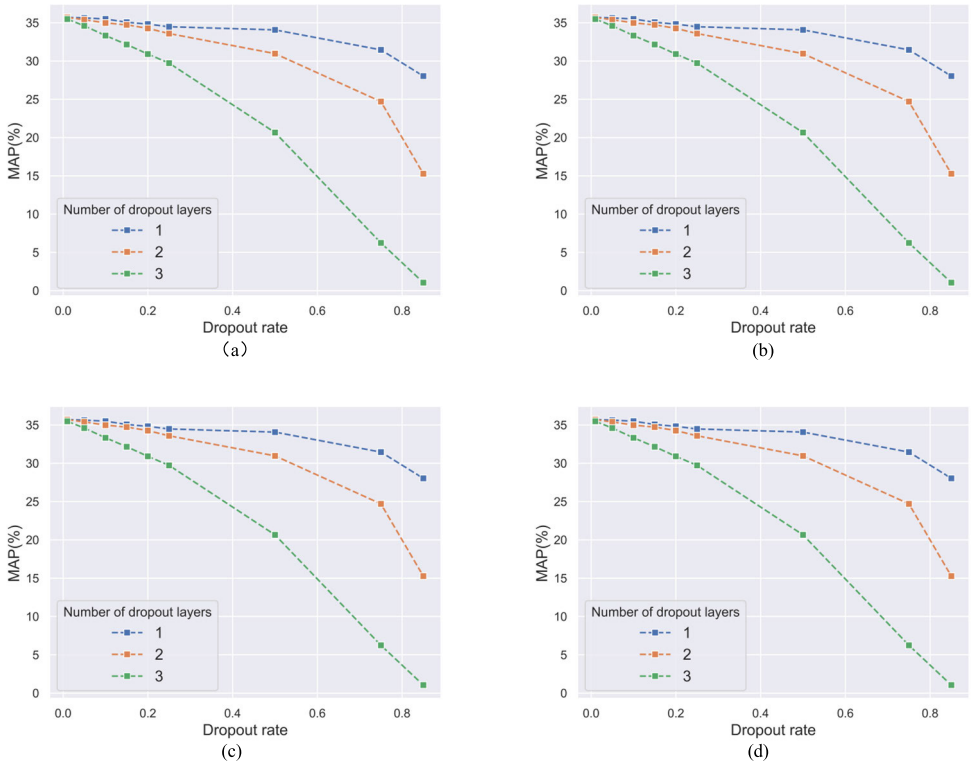


FIGURE 6. Sensitivity analysis of dropout layers and dropout probability, a MAP, b PDQ, c Avg_label and d Avg_spatial at MS COCO.

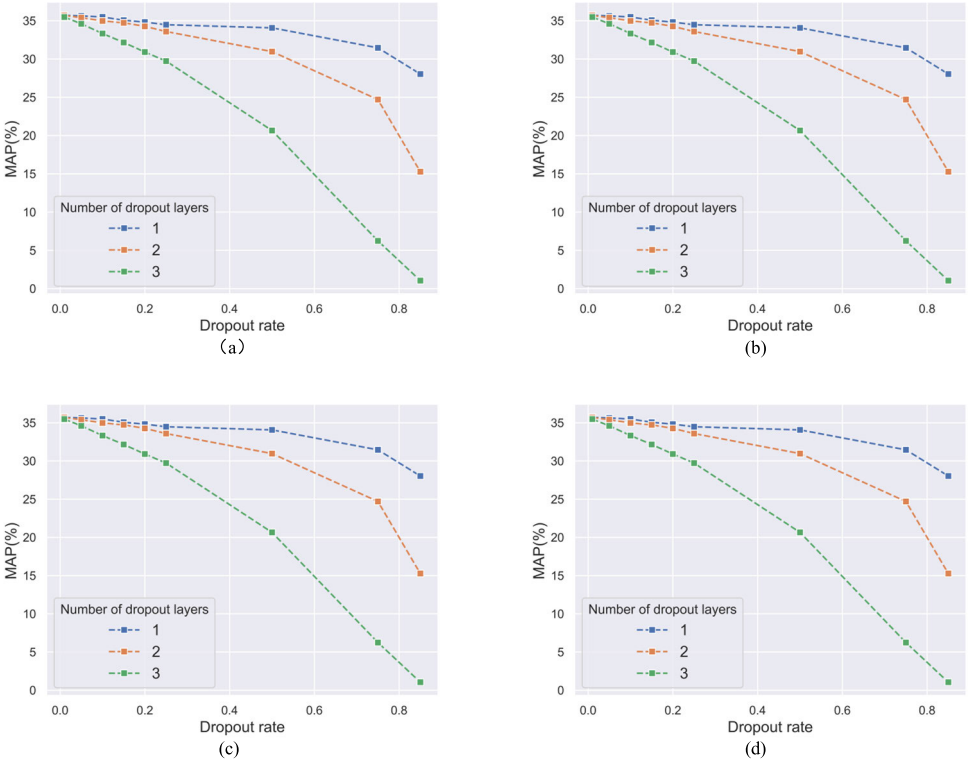


FIGURE 7. Sensitivity analysis of dropout layers and dropout probability, a MAP, b PDQ, c Avg_label and d Avg_spatial at BDD100K.

of Fig. 4(a), 4(b), 4(c), and 4(d) represent the MAP, PDQ, *Avg_label*, and *Avg_spatial* scores, respectively.

From Fig. 4, one can observe that the three curves share the same trend. This is because the effect of p on the rating indicators has a low correlation with the effect of the added Dropout position on the evaluation indicators, meaning the size of p does not affect the optimal position for adding the Dropout layer. Apart from this, MAP and PDQ have negatively correlated features, but a more precise detector can achieve higher MAP and PDQ scores simultaneously. This is because the randomness introduced by Dropout affects the quality of object detection, and the introduced randomness is the source of uncertainty prediction. When evaluating with PDQ, better scores appear at positions 17, 18, 21, while scores at positions 16, 19, 22, 24 drop significantly. This is because positions 17, 18, 21 are characterized by being in the middle layers of the detection head and located after Concat or C3 modules. In contrast, positions 16, 19, 22, 24 are characterized by being at the convolution modules, subsampling modules, or the end of the detection head, where the Dropout layer has a smaller impact on the convolution layer. Therefore, adding Dropout before the convolution layer is a better MC-Dropout solution. The label quality trend aligns with the MAP score, and the spatial quality trend aligns with the PDQ score, indicating that label quality has a high correlation with the MAP indicator, while spatial quality has a better correlation with the PDQ evaluation indicator.

Fig. 6 illustrates the sensitivity analysis results concerning Dropout probability and the number of Dropout layers. Each plot contains three curves corresponding to different numbers of Dropout layers n: when $n = 1$, a Dropout layer is added after the first detection head's C3 module; when $n = 2$, Dropout layers are added after the first and second detection heads' C3 modules; when $n= 3$, Dropout layers are added after the C3 modules of the three detection heads. The horizontal axis represents the Dropout probability, while the vertical axes of Fig. 6 (a), (b), (c), and (d) indicate the MAP, PDQ, *Avg_label*, and *Avg_spatial* scores, respectively.

From Fig. 5, it can be observed that the three PDQ curves exhibit a trend of initially increasing and then decreasing, with the peak of the curves gradually shifting forward as the number of Dropout layers increases. This is because there is a certain correlation between the number of Dropout layers and Dropout probability; increasing either can enhance randomness. The three curves for the spatial quality indicator do not follow a similar trend. This is because an excessive number of Dropout layers and a large Dropout probability will cause irreversible damage to detection quality, and thus, a higher number of Dropout layers should not be paired with an excessively high Dropout probability. The MAP and label quality curves share the same trend: as the Dropout rate and the number of Dropout layers increase, the quality gradually decreases. This is because a high level of randomness can cause a certain degree of disruption to the features extracted by the neural network.

TABLE 1. Ablation experiment of MC-drop.

	MAP (%)	PDQ (%)
M-YOLOv5s	25.69	21.59
M-YOLOv5m	33.93	29.39
M-YOLOv5l	37.91	32.17
M-YOLOv5x	40.32	33.24
YOLOv5s [42]	28.97	12.35
YOLOv5m [42]	36.15	20.93
YOLOv5l [42]	40.56	23.38
YOLOv5x [42]	44.37	24.63

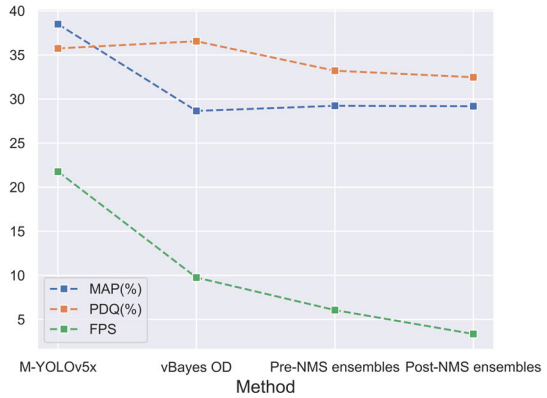


FIGURE 8. Performance comparison of models targeting the BDD100K dataset.

Fig. 5 and Fig. 7 represent sensitivity analyses conducted on the BDD100K dataset, with experimental settings identical to those used for the MS COCO dataset, except for the change in dataset. This was done to verify that the above conclusions are not unique to the MS COCO dataset. From the figures, it is evident that the characteristics exhibited are similar to those of the MS COCO dataset.

IV. EXPERIMENTS

This section first introduces the experimental environment configuration and then compares the performance of the M-YOLOv5 model with typical probabilistic object detection models, highlighting the superiority of the M-YOLOv5 model. Finally, the detection results of the M-YOLOv5 algorithm on some edge-case scenarios are presented.

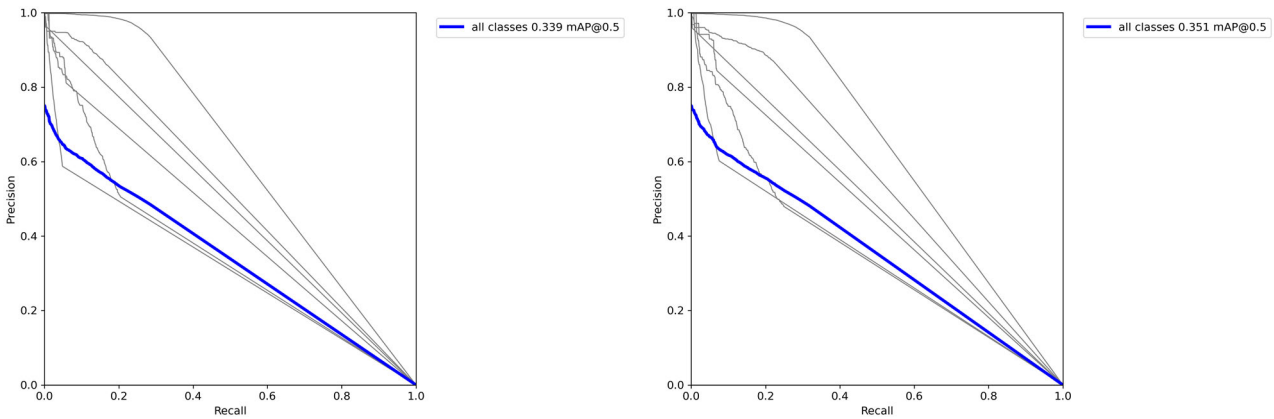
A. EXPERIMENTAL ENVIRONMENT CONFIGURATION

The experiment used an i7-13700fk CPU and a P100 16G GPU as the hardware system, with Ubuntu 20.04 as the operating system. The M-YOLOv5 model was built based on Python tools and accelerated using CUDA 11.7.

This paper selects the MS COCO 2017 dataset as the training and validation dataset, which includes 118,287 and 5,000

TABLE 2. A summary of performance comparison between state-of-the-art probabilistic object detectors and M-YOLOv5 at MS COCO.

Method	MAP (%)	PDQ (%)	pPDQ (%)	Sp (%)	Lbl (%)	FG (%)	BG (%)	TP	FP	FN	FPS
M-YOLOv5s (0.5)	25.69	21.59	52.55	45.14	71.14	74.64	65.78	15788	1655	20993	104.17
M-YOLOv5m (0.5)	33.93	29.39	54.34	45.86	74.41	74.11	67.36	18436	2407	13237	86.23
M-YOLOv5l (0.5)	37.91	32.17	55.61	46.73	76.49	74.81	67.75	20011	2922	11662	56.71
M-YOLOv5x (0.5)	40.32	33.24	55.97	46.78	77.70	75.26	67.42	20805	3354	10868	21.76
probFRCNN(0.5) [39]	35.5	28.4	56.7	45.0	90.7	77.8	60.7	23434	10016	13347	9.73
MC-Dropout SSD (0.5) [43]	15.8	12.8	47.3	39.9	74.0	73.1	57.3	10510	2165	26271	43.96
MC-Dropout SSD (0.05) [43]	19.5	1.3	26.1	27.3	35.9	60.1	46.2	24843	461074	11938	43.96
SSD-300 (0.5) [44]	15.0	3.9	18.1	9.7	80.2	57.5	25.1	8999	4746	27782	47.12
SSD-300 (0.05) [44]	19.3	0.6	9.7	6.4	40.2	38.1	32.3	21961	324067	14820	47.12
YOLOv3 (0.5) [45]	29.7	5.7	14.6	6.2	95.8	52.2	20.4	17390	7728	19391	52.14
YOLOv3 (0.05) [45]	30.1	3.3	12.2	5.1	92.8	44.6	22.9	23447	50074	13334	52.14
FRCNN R (0.5) [46]	32.8	6.7	19.1	10.3	88.8	62.2	23.6	19930	20044	16851	7.63
FRCNN R (0.05) [46]	34.3	3.0	17.1	9.5	78.5	57.8	25.1	23081	93141	13700	7.63
FRCNN R+FPN (0.5) [47]	34.6	11.8	27.1	16.9	86.5	60.6	35.7	22537	14706	14244	11.47
FRCNN R+FPN (0.05) [47]	37.0	4.2	23.1	15.8	69.5	54.4	38.7	29326	123511	7455	11.47
FRCNN X+FPN (0.5) [47]	37.4	11.9	27.9	17.6	88.2	60.8	36.8	24523	20444	12258	8.36
FRCNN X+FPN (0.05) [47]	39.0	4.4	24.8	16.7	74.4	55.6	39.1	29922	130009	6859	8.36



MS COCO BDD100K

FIGURE 9. Comparison of P-R curves between MS COCO dataset and BDD100K dataset.

images, respectively. In line with the edge-case standards proposed in ISO 21448 [40] and by Bogdoll [41], 20 groups of edge-case scenarios are selected from the MS COCO dataset for testing, and a set of results containing all types of edge cases is chosen for presentation. The model also underwent training and validation on the BDD100K dataset, comprising 70,000 training images and 10,000 testing images, primarily captured on roads and serving as a real-world dataset for autonomous driving training. During experimentation, sampling is performed ten times, with a label confidence threshold of 0.5 and an IoU confidence threshold of $u = 0.6$. In addition, the model parameters are the same as those of the original model. The weights used to detect edge cases are obtained by training for 30 rounds on the basis of YOLOv5 pre-trained weights, with each round of training the YOLOv5 model requiring 50 minutes. Conducting

FIGURE 10. Comparison of running time of uncertainty modeling methods.

comparative experiments using deep ensemble methods, the weights are trained ten times using YOLOv5 pre-trained weights with different data augmentation techniques, with each 30-round training session yielding ten different weights.

B. ABLATION EXPERIMENT OF MC-DROP

In order to investigate the performance improvement of the embedded MC-drop method on the YOLOv5 algorithm, this study conducted ablation experiments on four models of the YOLOv5 algorithm. The results are shown in Table 1. Based on the results of sensitivity analysis, a comparative experiment was carried out using a combination of a favorable 25% dropout rate and a single dropout. In terms of conventional evaluation metrics, there is a slight decline in the performance of the four M-YOLOv5 detection algorithms compared to the original YOLOv5 algorithm. This is because the MC-drop method randomly drops some neurons. Despite the low probability of dropout and its embedding in layers that don't affect feature extraction, the discarding of some key neurons is inevitable. From the perspective of the PDQ metric, the scores of the four M-YOLOv5 algorithms show a significant improvement compared to YOLOv5. This demonstrates the pronounced effect of the MC-drop method in enhancing the probability quality of the detector. The M-YOLOv5 algorithm significantly enhances the quality of detection probability in the results without sacrificing detection accuracy, thereby strengthening its safety and robustness for use in autonomous driving.

C. MODEL PERFORMANCE COMPARISON

To demonstrate the superiority of the M-YOLOv5 model, this paper compares it with mainstream probabilistic object detection models targeting the MS COCO dataset, such as probFRCNN by Hall et al. [39], and MC-Drop SSD by Miller et al. [43]. Additionally, the same uncertainty modeling method as M-YOLOv5 was applied to object detectors like SSD-300 by Liu et al. [44], YOLOv3 by Redmon and Farhadi [45], the FRCNN series by Yang et al. [46], and the FRCNN+FPN series by Massa and Girshick [47], followed by performance testing. The comparative results are presented in Table 2. The paper applies the proposed MC-drop method to the four varying network depths of YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These modified models are respectively named M-YOLOv5s, M-YOLOv5m, M-YOLOv5l, and M-YOLOv5x.

The aim of modeling uncertainty in the detection results of the target detection model is to maintain high uncertainty evaluation quality without significantly compromising detection accuracy. MAP is the quantification metric for detection accuracy, while Probability-based PDQ is the metric for uncertainty evaluation quality. The factors for calculating MAP and PDQ include pPDQ, Sp, Lbl, FG, BG, TP, FP, and FN. Additionally, considering the application scenario of high real-time safety-critical applications, this paper compares the real-time performance of all methods. Table 2 presents the comparison results for all methods in terms of the detection accuracy metric MAP, the uncertainty evaluation quality metric PDQ, and real-time performance. It also lists the values of factors pPDQ, Sp, Lbl, FG, BG, TP, FP, and FN used in the calculation of MAP and PDQ.

The first four rows of the table represent the M-YOLOv5 models of varying depths, with M-YOLOv5s, M-YOLOv5m, M-YOLOv5l, and M-YOLOv5x having progressively increasing model depths. The table shows that with increasing model depth, both MAP and PDQ scores gradually increase, but FPS decreases. This implies that deeper models have higher detection accuracy and uncertainty evaluation quality but poorer real-time performance. Compared to the current state-of-the-art probabilistic object detection models, our proposed M-YOLOv5x model demonstrates the best performance in both MAP and PDQ, with a detection speed of 21.76 frames/s. The probFRCNN algorithm has a similar but slightly lower accuracy performance; however, its detection speed is 55.28% lower than that of M-YOLOv5x. Models like MC-Dropout SSD, SSD-300, and YOLOv3, which have detection accuracy, as indicated by their lower MAP and PDQ scores, making them unsuitable for practical applications in the field of autonomous driving. higher detection speeds, achieve this at the significant cost of compared to the best performance among these three algorithms, our M-YOLOv5l model shows an 8.76% increase in detection speed, along with a 15.58% increase in MAP score and a 151.32% increase in PDQ score, thereby proving that the model proposed in this work has the optimal overall performance.

Furthermore, the performance of the M-YOLOv5x model was compared with the advanced Bayes OD probabilistic object detection algorithm [37] targeting the BDD100K dataset, demonstrating the versatility of the M-YOLOv5 algorithm. The experimental results are illustrated in Fig. 8. To visually represent the variation in key metrics among different algorithms, the MAP, PDQ, and FPS results of the four algorithms are depicted in a line graph. It is evident from the graph that the M-YOLOv5 algorithm, with minimal loss in PDQ accuracy, shows significant improvements in MAP and FPS, meeting the real-time requirements of autonomous driving. The comparison of the P-R curves for the MS COCO and BDD100K datasets is shown in Fig. 9.

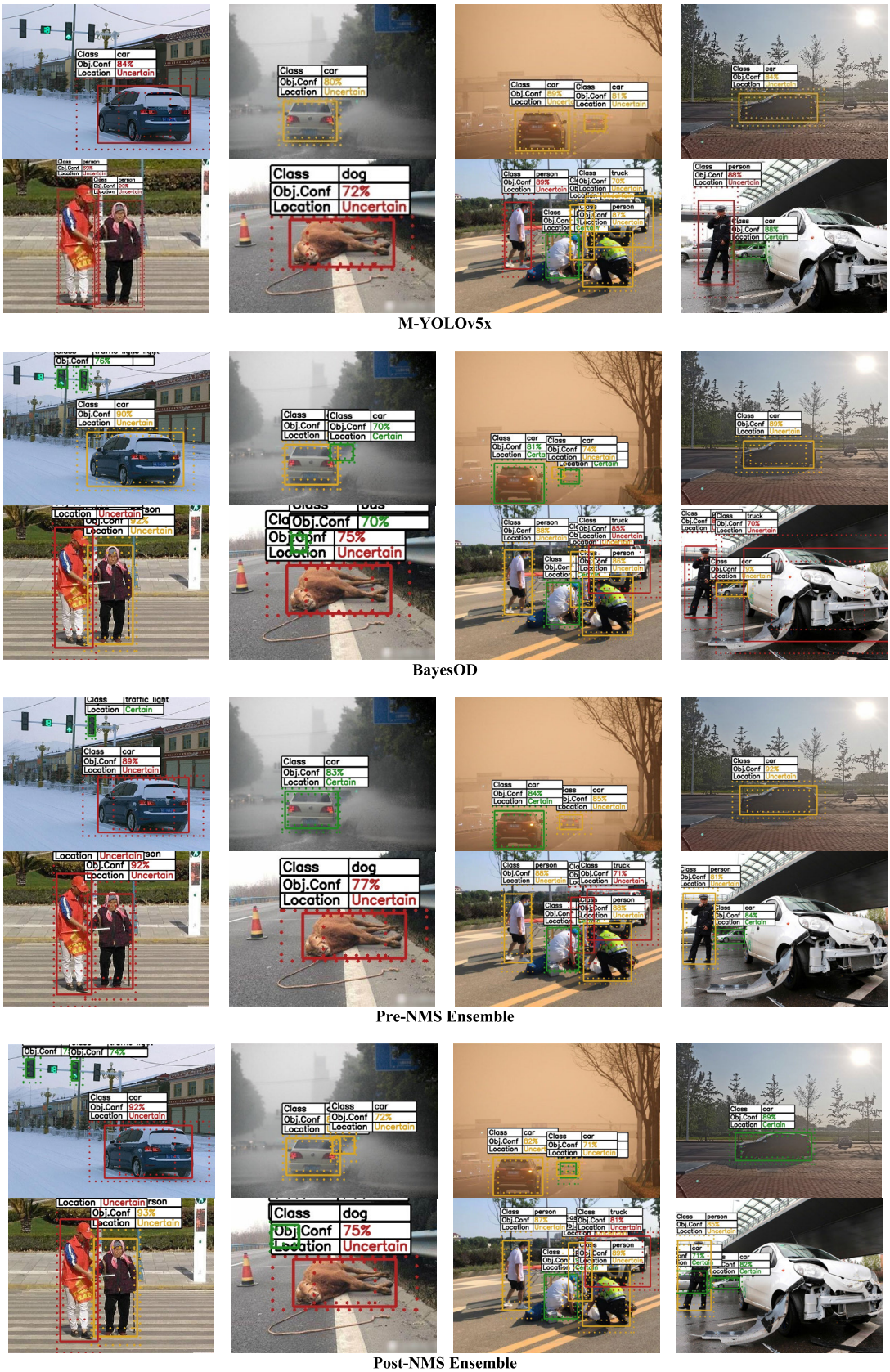


FIGURE 11. Visual comparison of M-YOLOv5 with current advanced probabilistic object detection algorithms in edge case scenarios.

Additionally, using YOLOv5m as an example, this paper compares the time required for the MC-Drop method in optimizing the uncertainty quantification process with the Ensemble method, as depicted in Fig. 10. The experiments were conducted using a P100 16G GPU, where YOLOv5m represents the original YOLOv5m model, M-YOLO represents the uncertainty modeling using the MC-Drop method with ten samplings, Ensemble-5 represents the ensemble of five weights, and Ensemble-10 represents the ensemble of ten weights. In Fig. 10, the horizontal axis represents the algorithm, and the vertical axis represents the time required to detect a single image. Each algorithm was tested twenty times in the experiments, and the size of the box in the boxplot represents the fluctuation in detection time. It can be inferred from the figure that compared to the original YOLOv5 model without uncertainty quantification, M-YOLOv5 shows a slight increase in detection time. However, it has a shorter detection time compared to the ensemble method with five weights, and its detection time is significantly lower than that of the ensemble method with ten weights. This advantage is attributed to the optimization of the MC-Drop uncertainty quantification process. Furthermore, the M-YOLOv5 model exhibits the smallest time fluctuation, demonstrating the best robustness of the M-YOLOv5 algorithm.

Finally, we compared our M-YOLOv5 with advanced object detection algorithms tailored for the MS COCO dataset, achieving a reduction in spatial complexity by 25.6% and in temporal complexity by 53.2%. Additionally, when against probabilistic object detection algorithms designed for the BDD100K dataset, the M-YOLOv5 demonstrated superior performance, reducing spatial complexity by 27.6% and temporal complexity by 55.3%, thereby affirming the exceptional capabilities of the M-YOLOv5 model.

D. M-YOLOv5 CORNER CASE TEST

This paper employs the M-YOLOv5 model to test some edge-case scenarios within the MS COCO dataset, finding that in comparison to regular conditions, our model can offer higher spatial uncertainty in object detection within these scenarios. We conducted a total of twenty test groups, and the tests indicate that the uncertainty quality of the M-YOLOv5 model is higher. We chose a test set including extreme weather, natural disasters, abnormal lighting, with the results shown in Fig. 11. It can be observed that in these edge-case scenarios, the predictive confidence of the M-YOLOv5 model is relatively low, indicating that the detection results are unreliable, and necessitating corresponding behavior from the decision-making layer to ensure the safety of autonomous vehicle operation. Compared to object detection models without uncertainty estimation, probabilistic object detection models, in these cases, allow the decision system to recognize the insufficiency of the reliability in the perception system's output. This understanding enables the implementation of conservative safety measures to avoid collisions.

As shown in Fig. 11, we visualized the model detection results of BayesOD, Pre-NMS Ensemble and Post-NMS Ensemble. To facilitate the comparison of these visualizations, we standardized the format of various algorithms to match our own, selecting the outcomes derived from their models accordingly. The images reveal that the M-YOLOv5 algorithm possesses superior quality of uncertainty in adverse weather conditions and with abnormal traffic participants. For instance, in each algorithm's second image, the vision is extremely blurred due to heavy rain, leading to M-YOLOv5's uncertainty regarding the detected object's location, whereas the Pre-NMS Ensemble algorithm is very confident in its detection result. Similarly, in the fourth image, M-YOLOv5 remains uncertain about its detection outcome, while Post-NMS Ensemble is highly confident in its result. Overconfidence in detection results under extreme conditions can pose a threat to the safety of autonomous driving.

V. CONCLUSION AND FUTURE DIRECTIONS

This research systematically introduces the M-YOLOv5 model, an extension of the YOLOv5 object detection algorithm with uncertainty modeling using the MC-Drop method. Sensitivity analysis of hyperparameters that significantly impact MC-Drop was conducted, shedding light on the intricate relationship between the Dropout layers and detection quality. Recognizing the limitations of the MAP evaluation metric, the study also incorporates PDQ, offering a more comprehensive evaluation system. Performance comparisons with leading probabilistic object detection models highlight the superiority of the M-YOLOv5 algorithm. The research represents a significant step in advancing probabilistic object detection, delivering both enhanced performance and valuable insights into modeling uncertainty, demonstrating the advantages of the M-YOLOv5 model for applications demanding reliability and efficiency, such as autonomous driving.

However, there is still significant room for improvement in the detection speed, detection progress, and uncertainty prediction quality of the M-YOLOv5 method. In the future, we plan to continue optimizing the operation mechanism of MC-drop to reduce the prediction time of the probabilistic object detection model. In addition, current probabilistic object detection algorithms can only model the uncertainty of detection results as a whole, without being able to ascertain the extent to which different sources of noise contribute to this uncertainty. For instance, M-YOLOv5 can detect the combined impact of weather conditions, sensor accuracy, and data annotation on the uncertainty of detection results, but it cannot determine which of these factors has the most significant impact. Moving forward, we will explore how to decompose and quantify the individual contributions of different sources of uncertainty, which will aid in improving the detector's performance and enhancing the interpretability of detection results.

CONFLICT OF INTEREST

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

REFERENCES

[1] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection," in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 95–101.

[2] Z. Wu, C. Liu, C. Huang, J. Wen, and Y. Xu, "Deep object detection with example attribute based prediction modulation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 2020–2024.

[3] M. Andronie, G. Lăzăroiu, M. Iatagan, I. Hurloiu, R. Ştefănescu, A. Dijmărescu, and I. Dijmărescu, "Big data management algorithms, deep learning-based object detection technologies, and geospatial simulation and sensor fusion tools in the Internet of Robotic Things," *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 2, p. 35, Jan. 2023.

[4] O. Zohar, K.-C. Wang, and S. Yeung, "PROB: Probabilistic objectness for open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 11444–11453.

[5] A. M. Roy, R. Bose, and J. Bhaduri, "A fast accurate fine-grain object detection model based on YOLOv4 deep neural network," *Neural Comput. Appl.*, vol. 34, pp. 3895–3921, Jan. 2022.

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 650–657, doi: 10.1109/FG.2017.82.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1137–1149.

[10] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," 2023, *arXiv:2304.00501*.

[11] X. Jia, Y. Tong, H. Qiao, M. Li, J. Tong, and B. Liang, "Fast and accurate object detector for autonomous driving based on improved YOLOv5," *Sci. Rep.*, vol. 13, no. 1, p. 9711, Jun. 2023.

[12] D. Snegireva and G. Kataev, "Vehicle classification application on video using YOLOv5 architecture," in *Proc. Int. Russian Autom. Conf.*, Sep. 2021, pp. 1008–1013.

[13] Y. Huang and H. Zhang, "A safety vehicle detection mechanism based on YOLOv5," in *Proc. IEEE 6th Int. Conf. Smart Cloud (SmartCloud)*, Nov. 2021, pp. 1–6.

[14] X. Song and W. Gu, "Multi-objective real-time vehicle detection method based on YOLOv5," in *Proc. Int. Symp. Artif. Intell. Appl. Media*, May 2021, pp. 142–145.

[15] H. F. Le, L. J. Zhang, and Y. X. Liu, "Surface defect detection of industrial parts based on YOLOv5," *IEEE Access*, vol. 10, pp. 130784–130794, 2022.

[16] F. Sun, Z. Li, and Z. Li, "A traffic flow detection system based on YOLOv5," in *Proc. 2nd Int. Seminar Artif. Intell., Netw. Inf. Technol. (AINIT)*, 2021, pp. 458–464.

[17] Y. Huo, J. Zhang, X. Du, X. Wang, J. Liu, and L. Liu, "Recognition of parasite eggs in microscopic medical images based on YOLOv5," in *Proc. 5th Asian Conf. Artif. Intell. Technol. (ACAIT)*, Oct. 2021, pp. 123–127.

[18] D. Chen, D. Miao, and X. Zhao, "Hyneter: Hybrid network transformer for object detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[19] G. Qi, X. Zhang, and X. Fu, "MT-YOLO: Combination of multi-scale feature extraction and transformer in one-stage object detection," in *Proc. Int. Seminar Comput. Sci. Eng. Technol. (SCSET)*, 2023, pp. 314–317.

[20] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3D LiDAR-based video object detection for autonomous driving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2068–2078, Apr. 2022.

[21] S. Thomas and K. M. Groth, "Toward a hybrid causal framework for autonomous vehicle safety analysis," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 237, no. 2, pp. 367–388, Apr. 2023.

[22] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, Mar. 2023.

[23] J. Tellinghuisen, "Statistical error propagation," *J. Phys. Chem. A*, vol. 105, no. 15, pp. 3917–3921, Apr. 2001.

[24] M. T. Le, F. Diehl, T. Brunner, and A. Knoll, "Uncertainty estimation for deep neural object detectors in safety-critical applications," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3873–3878.

[25] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time LiDAR 3D object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1280–1287.

[26] D. Feng, Y. Cao, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging uncertainties for deep multi-modal object detection in autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 877–884.

[27] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2888–2897.

[28] J. Guo and S. Gould, "Deep CNN ensemble with data augmentation for object detection," 2015, *arXiv:1506.07224*.

[29] H. Lin, C. Sun, and Y. Liu, "OBBStacking: An ensemble method for remote sensing object detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2112–2120, 2023, doi: 10.1109/JSTARS.2023.3243168.

[30] J. Lee, S.-K. Lee, and S.-I. Yang, "An ensemble method of CNN models for object detection," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 898–901.

[31] S. H. Yelleni and D. Kumari, "Monte Carlo DropBlock for modeling uncertainty in object detection," *Pattern Recognit.*, vol. 146, Feb. 2024, Art. no. 110003.

[32] R. Seoh, "Qualitative analysis of Monte Carlo dropout," 2020, *arXiv:2007.01720*.

[33] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3243–3249.

[34] D. Miller, N. Sünderhauf, H. Zhang, D. Hall, and F. Dayoub, "Benchmarking sampling-based probabilistic object detectors," in *Proc. CVPR Workshops*, vol. 3, 2019, p. 6.

[35] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOV3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 502–511.

[36] T. Azevedo, R. de Jong, M. Mattina, and P. Maji, "Stochastic-YOLO: Efficient probabilistic object detection under dataset shifts," 2020, *arXiv:2009.02967*.

[37] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022.

[38] J. Xiao, H. Guo, J. Zhou, T. Zhao, Q. Yu, Y. Chen, and Z. Wang, "Tiny object detection with context enhancement and feature purification," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118665.

[39] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Suenderhauf, "Probabilistic object detection: Definition and evaluation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jun. 2020, pp. 1020–1029.

[40] *Road Vehicles—Safety of the Intended Functionality (DIS)*, Standard ISO 21448, 2020.

[41] D. Bogdoll, J. Breitenstein, F. Heidecker, M. Bieshaar, B. Sick, T. Fingscheidt, and M. Zöllner, "Description of corner cases in automated driving: Goals and challenges," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1023–1028.

[42] G. Jocher. (2020). *YOLOv5 by Ultralytics*. Accessed: Aug. 30, 2023. [Online]. Available: https://github.com/ultralytics/yolov5

[43] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf, "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2348–2354.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. Berg, "SSD: Single shot MultiBox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, 2016, pp. 21–37.

[45] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[46] J. Yang, J. Lu, D. Batre, and D. Parikh, "A faster PyTorch implementation of faster R-CNN," 2017. [Online]. Available: https://github.com/jwyang/faster-rcnn.pytorch
[47] F. Massa and R. Girshick, "MaskrCNN-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in PyTorch," 2018. [Online]. Available: https://github.com/facebookresearch/maskrcnn-benchmark

RUI ZHAO (Member, IEEE) was born in Liaoyuan, Jilin, China, in 1986. She received the B.S. degree in computer science and technology from Northeast Normal University, in 2009, and the Ph.D. degree in computer science and technology from Jilin University, Changchun, China, in 2017. She is currently an Associate Professor with the College of Automotive Engineering, Jilin University. She has authored about 30 journal articles and ten patents in China. Her research interests include cooperative control, functional safety, cybersecurity, and safety reinforcement learning for connected and automated vehicles.

KUI WANG was born in Zhoukou, Henan, China, in 2002. He received the B.E. degree in automotive engineering from Jilin University. He will study the M.E. degree with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China. His research interests include safety reinforcement learning and decision-making of autonomous vehicle.

YANG XIAO was born in Tangshan, Hebei, China, in 1984. He received the Ph.D. degree in solid mechanics and manufacturing engineering from Inha University. He is currently an Associate Professor with the College of Automotive Engineering, Jilin University. He published more than 20 journal articles, including more than ten SCI and EI papers. His research interests include the safety prediction and prevention strategy of new energy vehicle batteries and low speed autonomous driving. He is the Guest Editor of *Energies*.

FEI GAO received the B.S. and Ph.D. degrees in automotive engineering from Jilin University, Changchun, China, in 2011 and 2017, respectively. From 2014 to 2015, she was a Visiting Student in Berkeley, CA, USA. She is currently an Associate Professor with the State Key Laboratory of Automotive Simulation and Control Automotive Engineering, Jilin University. She is the coauthor of three books, more than 20 articles, and more than ten inventions. Her research interests include automotive human engineering and motion sickness.

ZHENHAI GAO (Member, IEEE) was born in Changchun, Jilin, China, in 1973. He received the Ph.D. degree in automotive engineering from Jilin University. He is currently the Deputy Dean of automotive engineering and the Director of the State Key Laboratory of Automotive Simulation and Control Automotive Engineering, Jilin University. He is the coauthor of three books. More than 100 articles have been published and 20 invention patents have been authorized. His research interests include autopilot technology and human engineering. He is a Distinguished Member of the Expert Committee Intelligent Connected Vehicle Innovation Alliance, the Chairperson of the Industrial Design Association in Jilin Province, and an Editorial Board Member of *International Journal of Human Factors Modelling and Simulation*.