Real Time Vehicle Detection from Captured Image

Mrs.KOKILA S M.E,
Assistant Professor,
Department of Computer Science And
Engineering,
Shree Venkateshwara HiTech Engineering
College,
Gobichettipalayam.
E-mail:kokiyelloever@gmail.com

Ms.DIVYA S,
Student,
Department of Computer Science And
Engineering,
Shree Venkateshwara Hi Tech Engineering
College,
Gobichettipalayam.
E-mail:divyas25052002@gmail.com

Ms.GOMATHI D,
Student,
Department of Computer Science And
Engineering,
Shree Venkateshwara Hi Tech Engineering
College,
Gobichettipalayam.
E-mail:gomathidhapal2003@gmail.com

Ms.LALITHA K S,
Student,
Department of Computer Science And Engineering,
Shree Venkateshwara Hi Tech Engineering
College,
Gobichettipalayam.
E-mail:lalitha.kolappalur@gamil.com

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-ofthe-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.

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 [1-3]. 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 lower halfdepicts extreme weatherconditions, 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 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.

II. M-YOLOV5 MODEL

*This section describes the proposed probabilistic object detection algorithm M-YOLOv5, which employs the MCDrop 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.

I. EASE OF USE

I. 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 $T = \{d_i, r_i\}_{i=1}^{N}$, where d_i is randomly selected input image data from the set D, and $r_i = +r_{,\%}, r_{,\&}, r_{,}$ corresponds to the target output data from the object detection result set R. Here, $r_{,\%}$ represents the type of object and the probability of each class, $r_{,\&}$ represents the position of the object in the image, and r denotes the uncertainty of the detection result. Let $c \in \{1,2, \dots, C\}$ represent the category code corresponding to the target, where C is the total number of target classes. Let $i \in \{1,2, \dots, I\}$ indicates the current number of samples, where I represents the sampling times of the object detector. Let $obj_i \in \{1, \dots, C\}$ represents the class of the object, and the probability of each class. Then $p_{)(}, p_{\#(}, \dots, p$ define $*($ respectively represent $r_{,\%} = \sum \#_{\$-\%\_!}$", where $r_{\%(} = (obj_(, p)(, p\#(, \dots, p*()$. Let $x_(, y_($ represent the coordinates of the center of the predicted box, and $h_($ the height of the box, th$^(= (x_(, y_(, w_(, h_()$. Define en $r_{,}$ is expressed as $p_{\%}(r_{,\%}|dw, D_((r,)$the width, $_, =$ as the $\sum \#\$\_\%\_\&$",

where $r_{\&}$probability that the input data d leads to an object

class of $r_{,\%}$ under a specific object detection model, and $p_{\&}(r_{,\&}|d, D)$ as the probability that the object location is $r_{,\&}$, then $r^($ is expressed as $r^(= Dp_{\%}(r_{,\%}|d, D), p_{\&}(r_{,\&}|d, D)E$. The goal of this paper is to provide an accurate estimate of object detection class uncertainty $p_{\%}(r_{\%}^(|d, D)$ and location uncertainty $p_{\&}Dr_{\&}^(|d, DE$, along with the detection class results

$r_{,\%}$ and location results $r_{,\&}$, based on the original object detection model f, by designing the MC-drop method, and according to the input image data $\{d_i\}_{i=1}^{N}$.
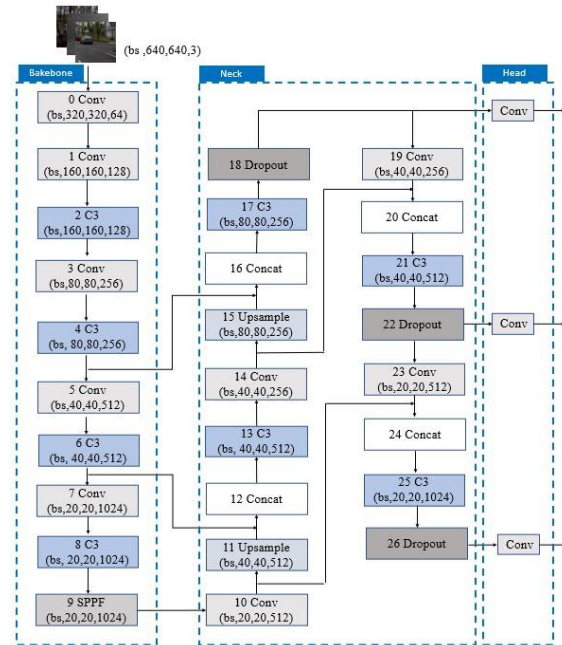
FIGURE 2.M-YOLOv5 model structure diagram.

II. 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 $d^)$ of the CNN, it first passes through the Backbone layer, resulting in the input $d_{!/\%0}$ to the Neck structure:

$$d_{!/\%0} = Backbone(d) \qquad (1)$$

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:

$$d!/\%0 = F(d)) = H0Dd01\#, H01\#(d012), \dots , H\#(d))E \quad (2)$$

where F represents the CNN network model, and H_0 is the operation function of the k-th layer in the network structure. To avoid gradient accumulation leading to the relearning of redundant information, the Backbone network structure
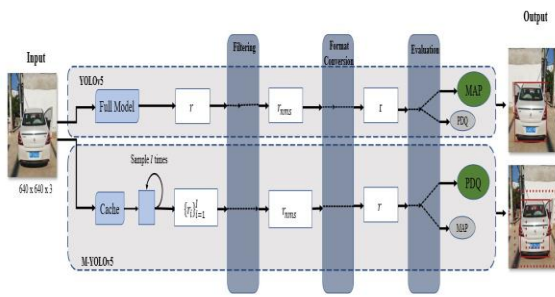
$$d0 = M \text{ PQ}d01\#\text{'}, T \text{ PFD}d01\#\text{''ERSR}$$

Here, $d_{31\#}$' and $d_{31\#}$'' are two parts of $d_{31\#}$ divided along the channel, T is a transition function truncating the gradient flow of $H_\#$, H_2, … ,H_0, 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

$d4,5$'567:

$$d4,5'567 = Neck(d!/\%0) \quad (4)$$

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,

first obtains $\{r_i\}^I_{i''1}$ 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).

The MC-drop method can approximate the posterior focuses optimization on each layer's network model $H_($, and distribution of Bayesian inference through the Dropout the output of the k-th layer is expressed as: method, and thereby quantify the uncertainty of the object detection model

$(3)d_{4,5'567}$ into the Dropout layer, the input $d_{8/94}$ to the 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.
detection head is obtained:

$$d8/94 = d4,5'567 *diag'm{:}(\text{a;:""} \# \quad (5)$$

$$m{:}^(\sim Bernoulli(p) \qquad for$$
$$j = 1, \dots ,K_(\qquad (6)$$

where $m{:}^($ 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.

A. 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 of Fig. 4(a), 4(b), 4(c), and 4(d) represent theMAP, 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.

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

B. 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 PostNMS Ensemble is highly confident in its result. Overconfidence in detection results under extreme conditions can pose a threat to the safety of autonomous driving.

C. III.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 morecomprehensive 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.

References

IV REFERENCES

[1] A. Womg, M. J. Shafiee, F. Li and B. Chwyl, "Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for RealTime Embedded Object Detection," *2018 15th Conference on Computer and Robot Vision (CRV),* Toronto, ON, Canada, 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," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Singapore, Singapore, 2022, pp. 20202024.

[3] M. Andronie et al., "Big Data Management Algorithms, Deep Learning-Based Object Detection Technologies, and Geospatial Simulation and Sensor Fusion Tools in the Internet of Robotic Things," *ISPRS International Journal of Geo-Information*, vol. 12, no. 2, 2023.

[4] O. Zohar, K. -C. Wang and S. Yeung, "PROB: Probabilistic Objectness for Open World Object Detection," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Vancouver, BC, Canada, 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 Computing and Applications,* pp. 1-27, 2022.

[6]. T Senthil Prakash, V CP, RB Dhumale, A Kiran., "Auto-metric graph neural network for paddy leaf disease classification" - Archives of Phytopathology and Plant Protection, 2023.

[7]. T Senthil Prakash, G Kannan, S Prabhakaran., "Deep convolutional spiking neural network fostered automatic detection and classification of breast cancer from mammography images" - Research on Biomedical Engineering,

[8]. TS Prakash, SP Patnayakuni, S Shibu., "Municipal Solid Waste Prediction using Tree Hierarchical Deep Convolutional Neural Network Optimized with Balancing Composite Motion Optimization Algorithm" - Journal of Experimental & Theoretical Artificial2023,

[9]R. Senthilkumar, B. G. Geetha, (2020), Asymmetric Key Blum-Goldwasser Cryptography for Cloud Services Communication Security, Journal of Internet Technology, vol. 21, no. 4 , pp. 929-939.

[10]Senthilkumar, R., et al. "Pearson Hashing B-Tree With Self Adaptive Random Key Elgamal Cryptography For Secured Data Storage And Communication In Cloud." Webology 18.5 (2021): 4481-4497

[11]Anusuya, D., R. Senthilkumar, and T. Senthil Prakash. "Evolutionary Feature Selection for big data processing using Map reduce and APSO." International Journal of Computational Research and Development (IJCRD) 1.2 (2017): 30-35.

[12] Farhanath, K., Owais Farooqui, and K. Asique. "Comparative Analysis of Deep Learning Models for PCB Defects Detection and Classification." Journal of Positive School Psychology 6.5 (2022).