IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 August 2024, accepted 9 September 2024, date of publication 12 September 2024, date of current version 23 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3459629

RESEARCH ARTICLE

Enhancing Object Detection in Dense Images: Adjustable Non-Maximum Suppression for Single-Class Detection

KYEONGMI NOH[1], SEUL KI HONG[2], STEPHEN MAKONIN[3], (Senior Member, IEEE), AND YONGKEUN LEE[2], (Senior Member, IEEE)

[1]Department of Information Communication Media Engineering, Graduate School of Nano it Design Fusion, Seoul National University of Science and Technology, Seoul 01811, South Korea
[2]Department of Semiconductor Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[3]Computational Sustainability Laboratory, School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 0A7, Canada

Corresponding author: Yongkeun Lee (yklee@seoultech.ac.kr)

This work was supported by Seoul National University of Science and Technology.

ABSTRACT Deep learning-based object detection technology often relies on non-maximum suppression (NMS) algorithms to eliminate redundant detections. However, the conventional NMS algorithm struggles with distinguishing between overlapping and small objects due to its simple constraints. While Soft-NMS offers a slight improvement in object detection performance, it still falls short in addressing this challenge. Our proposed solution, adjustable-NMS, represents a significant advancement. While performing comparably to NMS and Soft-NMS on less dense images where objects are easily countable, adjustable-NMS excels in scenarios with higher object density or smaller objects. In such cases, it outperforms both NMS and Soft-NMS, showcasing notably superior object detection capabilities. On average, the improvement achieved with adjustable-NMS reaches an impressive 33.3%. This demonstrates adjustable-NMS's efficacy in enhancing object detection accuracy, particularly in challenging environments characterized by dense scenes or diminutive objects.

INDEX TERMS Adjustable-NMS, intersection over union, NMS, object detection, soft-NMS, YOLO.

I. INTRODUCTION

Deep learning is being actively studied as computational power improves significantly with high-performing graphics processing units (GPUs). Object detection, a key application area of deep learning, has been reviewed and applied in various domains such as face recognition, autonomous driving, and manufacturing defect recognition. These applications involve predicting and identifying specific objects in photos or videos [1], [2], [3], [4], [5], [6], [7], [8]. Widely known object detection deep learning models include the Region-based Convolutional Neural Network (R-CNN) [9], Fast R-CNN [10], Faster R-CNN [11], You Only Look Once (YOLO) [12], and Mask R-CNN. Object detection not only classifies objects in an image but also locates them using bounding boxes.

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

Non-Maximum Suppression (NMS) is an algorithm commonly used in the post-processing stage of object detection to refine bounding box predictions [13], [14]. NMS selects the bounding box with the highest confidence score and removes surrounding boxes by comparing them with a predefined Intersection over Union (IoU) threshold to eliminate duplicate results and enhance performance. While NMS performs well when objects in images do not overlap significantly, its performance degrades in scenes with dense overlapping objects or partially visible objects [15], [16], [17], [18], [19], [20], [21].

To address these limitations of NMS, various techniques have been researched and developed. Soft Non-Maximum Suppression (Soft-NMS) [22] was introduced to address this issue by decreasing the confidence scores of overlapping bounding boxes rather than eliminating them entirely, thus improving detection in cluttered scenes. However, it has limitations. In scenarios with very high object density or

severe occlusions, the decay function might not effectively distinguish closely packed objects, leading to suboptimal detection performance. Weighted NMS (WNMS) [23] is another variant that assigns a weighted score to overlapping boxes instead of removing them, thereby maintaining more information. This method maintains more information from the detected boxes, potentially improving the accuracy of the final detections. Nevertheless, the effectiveness of WNMS is highly dependent on the chosen weighting strategy. This strategy may not be optimal for all types of scenes, especially those with varying object sizes and densities, thus limiting its general applicability. Adaptive NMS techniques [24] have also been explored, where the IoU threshold is dynamically adjusted based on the density of the objects in the scene. This method aims to optimize the suppression process by tailoring the IoU threshold to the local context of detected objects. While this approach holds promise, it can be computationally expensive and complex to implement. Additionally, the adaptation might not be responsive enough for real-time applications, posing a significant challenge for practical deployment.

Other methods include IoU-Net [25], which learns an optimal IoU threshold directly from data, and soft IoU [26],which integrates IoU prediction into the confidence score for more precise suppression decisions. IoU-Net integrates IoU prediction directly into the neural network, enabling the model to predict the IoU between each detected box and the ground truth. This predicted IoU is then used to refine the suppression process, potentially improving accuracy. However, IoU-Net requires additional training and computational resources. Its effectiveness is heavily dependent on the quality of the IoU predictions, which might not always be accurate, thereby limiting its practical utility. Soft IoU incorporates the IoU prediction into the confidence score, making suppression decisions based on a combination of confidence and predicted IoU. This integration aims to enhance the precision of the suppression process. Similar to IoU-Net, Soft IoU's success hinges on accurate IoU predictions. Inaccuracies in these predictions can lead to suboptimal suppression results, thereby limiting its overall effectiveness.

Relation-NMS [27] and Distance-IoU (DIoU) NMS [28] are advanced variants of Non-Maximum Suppression (NMS) that enhance traditional methods in distinct ways. Relation-NMS improves NMS by incorporating contextual relationships between detected objects. It uses a graph-based approach to model interactions among bounding boxes, which can enhance suppression accuracy in crowded scenes. However, this method is complex and computationally demanding. Its effectiveness depends on the quality of contextual information, and an overemphasis on context might lead to the suppression of valid detections that are close but distinct.

Distance-IoU (DIoU) NMS extends traditional IoU-based NMS by adding a distance metric between bounding box centers, aiming to improve localization accuracy. This method refines the suppression process by considering both overlap and center distance, enhancing object differentiation. Despite its benefits, DIoU NMS may struggle with small or irregularly shaped objects and introduces additional computational complexity. It does not account for contextual relationships, which could be crucial in some detection scenarios. In summary, Relation-NMS provides advanced context-aware suppression but comes with increased complexity, while DIoU NMS improves localization accuracy but may be less effective for certain object scales and lacks contextual consideration.

Despite these advanced techniques [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], optimizing the IoU and confidence score thresholds remains crucial for enhancing detection capabilities in high-density overlapping object scenarios. This article proposes a method to find the optimal IoU and Confidence Score thresholds to enhance object detection capabilities in the YOLO model. We introduce an algorithm that optimizes these thresholds using the probability score of object presence in the bounding boxes detected by the YOLO network. The performance of the proposed algorithm was evaluated by comparing it to traditional NMS and Soft NMS methods, demonstrating significant improvements. Our main contribution is an adjustable algorithm that outperforms the current state-of-the-art Soft-NMS algorithm for object recognition in very dense images.

The paper is organized as follows: Section II provides a review of object detection research and the existing YOLO and Soft-NMS algorithms. Section III describes the methodology used to develop the adjustable-NMS algorithm. Section IV presents the results of two experiments, followed by conclusions in Section V.

The performance of the proposed adjustable NMS algorithm was compared with traditional NMS and Soft NMS methods. Traditional NMS serves as the baseline in most object detection frameworks, while Soft NMS is a widely adopted variant that addresses some of the limitations of traditional NMS.

Comparing our method with other NMS variants, such as Weighted NMS, Adaptive NMS, IoU-Net, Soft IoU, Relation-NMS, and Distance-IoU (DIoU) NMS, presents additional challenges. These methods often introduce significant complexity and require specific adjustments or additional training data, complicating a direct comparison. For instance, Relation-NMS and DIoU NMS involve advanced features like relationship modeling and distance-based calculations, necessitating substantial modifications to the evaluation framework, which can make a straightforward comparison difficult.

Moreover, IoU-Net and Soft IoU come with their own sets of hyperparameters and integration requirements that may not be directly compatible with the YOLO framework used in this study. Implementing and tuning these variants to ensure a fair comparison would require considerable additional effort and

resources, potentially overshadowing the core contributions of our proposed method.

Therefore, our study focused on demonstrating the effectiveness of the adjustable NMS algorithm against the most commonly used and straightforward techniques. This approach ensures a clear and direct comparison, aligning with practical and resource constraints while effectively showcasing the benefits of our proposed method.

II. BACKGROUND

A. OBJECT DETECTION MODELS

Object detection can detect and identify multiple objects, such as people, objects, and scenes, in an image or video and determine their locations. Deep-learning algorithms used in Object Detection are rapidly evolving and improved significantly. Object recognition systems are classified into one-stage and two-stage techniques according to how objects are learned and recognized. One-stage's representative model is YOLO, and two-stage's techniques include R-CNN family Fast-RCNN and Faster-RCNN. R-CNN [9] consists of a Region Proposal Network (RPN) and an Image Classification Network. R-CNN creates a region with a set of similar pixels and then performs Region Proposal through Selective Search to create more than 2,000 bounding boxes. Each bounding box is passed through CNN to extract features, and classes are classified through the Support Vector Machine (SVM). After classification, the extracted bounding box is matched to the pixel of the original image through regression. R-CNN has a slow speed because it extracts features from each of the more than 2,000 bounding boxes created by Region Proposal.

The Fast R-CNN [10], [37] algorithm was proposed to improve the performance of R-CNN. Fast R-CNN is an algorithm that significantly reduces redundant operations and improves the speed by ten times over R-CNN by performing single feature value extraction via passing the entire image through a CNN. In addition, Softmax was used instead of SVM to improve the learning speed for the model. The speed is better than that of R-CNN and Region Proposal was performed in an external network.

Faster R-CNN [38] modified the Region Proposal Network to be performed within the CNN. The same feature map as Fast R-CNN is extracted, and the object is found by moving a window of a specific size without using Selective Search. Information on k anchor boxes is extracted through the sliding window, each anchor box is converted through RoI pooling, and classification is performed.

Mask R-CNN [39] extracts pixels in which an object exists by adding a fully connected layer (FCN) to the Image Classification Layer and the Binding Box Regression Layer based on Faster R-CNN. Accuracy was improved using RoI Align, which improved Roll Polling.

YOLO [12], [40] is a model that combines Faster R-CNN's Region Proposal Network and Classifier into one network. The entire image is used to create a bounding box where end-to-end training is possible in real-time.

B. YOLO VARIANTS FOR OBJECT DETECTION

You Only Look Once (YOLO) is an algorithm that allows for finding objects at once and is designed to detect objects in real-time by dividing images into areas without using sliding windows. YOLO shows a good performance in Object Detection by performing object detection and classification simultaneously using convolutional neural networks. YOLO makes it easy to detect real-time objects and various models have been created according to the purpose of use. Fig. 1 shows the system structure of YOLO.

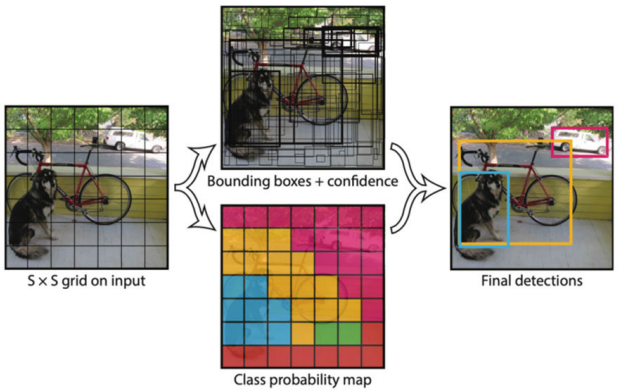


FIGURE 1. YOLO object recognition system.

YOLO divides the input image into $\mathcal{S}\times\mathcal{S}$ grid cells, and the cell closest to the center of the object among the divided grid cells detects the object. Each grid cell has $\mathcal{B}$ bounding boxes, and each bounding box has a probability score $\mathcal{S}$ of the existence of an object with the position coordinates. Intersection over Union (IoU), the overlapping ratio of the ground truth box and each bounding box, is calculated. YOLOv1 is fast, but it has a disadvantage of poor accuracy. YOLOv2 [41] normalizes the area and height of the ground truth bounding box and specifies the Anchor Box parameter using K-means clustering. In addition, various methods, such as batch normalization and high-resolution classifiers, are introduced to improve speed and accuracy compared to the existing YOLO. YOLOv3 [42] shows a good performance for accuracy, speed, and recognition of small objects by improving the problem of multiple small objects distributed in the narrow space of YOLOv2. YOLOv4 [43] uses CSPNet-based DarkNet53. The head consists of YOLOv3, and the neck consists of SPP and PANet. Using SPP, the size restriction of the input image is removed, so it increases the accuracy even with a large image size. Compared to the existing YOLOv3, the precision and accuracy have been increased. YOLOv5 [44] offers a straightforward, flexible framework implemented in PyTorch, achieving a good balance between detection speed and accuracy. YOLOv6 [45] enhances real-time object detection capabilities with improved speed and accuracy, featuring a refined network architecture for better feature extraction and object localization, and optimized training

processes for various hardware setups. YOLOv7 [46] incorporates novel techniques in network design and training strategies, excelling in detecting small objects and handling dense scenes with higher computational efficiency and enhanced robustness to varying conditions and object scales. YOLOv8 [47] achieves state-of-the-art performance in object detection with significant improvements in speed and accuracy, utilizing the latest advancements in deep learning and neural network architecture for optimal performance and versatility across a wide range of applications.

C. NMS AND SOFT-NMS ALGORITHMS

Non-maximum suppression (NMS) [22] merges predictive boxes adjacent to one object and is widely used as a post-processing step for the object detection framework.

YOLO divides each input image into $\mathcal{S}\times\mathcal{S}$ grid cells. Each divided grid cell has $\mathcal{B}$ bounding boxes, and each bounding box has a probability score $\mathcal{S}$ of the presence of an object with a range between 0 and 1 for the object class. By aligning the bounding box with the probability score $\mathcal{S}$ for the same object, the bounding box with the highest probability score $\mathcal{S}$ is set as the ground truth box, and the remaining bounding boxes are set as the prediction box. Intersection over Union (IoU) is used to calculate how much the ground truth box and prediction box overlap, indicating how much the actual value and the predictor match. Mathematically, $iou(\cdot)$ is expressed as (1) to calculate how much the ground truth box and prediction box overlap, indicating how much the actual value and the predictor match, as shown in Fig. 2.

$$iou = \frac{Ground\ Truth\ Box\ \cap Prediction\ Box}{Ground\ Truth\ Box\ \cup Prediction\ Box} \tag{1}$$

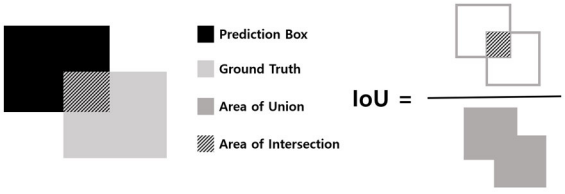


FIGURE 2. A visual definition of Intersection over Union (IoU).

In NMS, the IoU threshold N_t is a fixed bounding box larger than N_t is recognized as a duplicated bounding box, and the probability score $S_i = 0$ is set so that the bounding box is removed completely. A bounding box smaller than N_t is recognized as a box where another object exists, and the bounding box and the probability score s_i are maintained for further process.

However, in the NMS algorithm, when multiple objects of the same class are overlapped closely, the object detection performance is likely to be degraded by removing the correctly predicted bounding box by setting the probability score $s_i = 0$ as shown in (2). To improve the algorithm of NMS, a Soft-Non-Maximum Suppression (Soft-NMS) algorithm has been proposed. In Soft-NMS, when multiple objects of the same class are overlapped closely, the probability score is lowered but not zero, so that the bounding box and the probability score are maintained for further processing. Fig. 3 shows the object detection results of NMS and Soft-NMS.

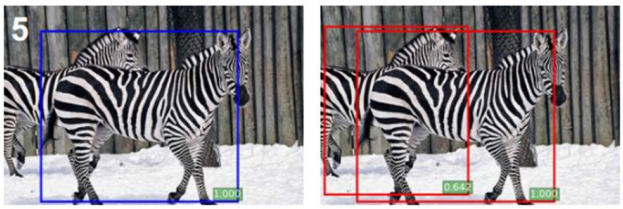


FIGURE 3. Object detection of NMS *vs.* Soft-NMS. Zebras detected: 1 *vs.* 2.

Soft-NMS sorts the detected bounding boxes in a high order of the probability score. The sorted bounding box has the number i. The bounding box with the highest probability score $\mathcal{S}$ is set to $\mathcal{M}$, and the IoU of $\mathcal{M}$ and the i-th bounding box was calculated. The probability score s_i is calculated according to (3).

$$s_i = \begin{cases} s_i, & \text{if } iou(\mathcal{M}, b_i) < N_t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$s_i = \begin{cases} s_i, & \text{if } iou(\mathcal{M}, b_i) < N_t \\ s_i(1 - iou(\mathcal{M}, b_i)), & \text{otherwise} \end{cases} \tag{3}$$

where s_i and $\mathcal{M}$ represent the probability score of the i-th bounding box and the bounding box with the highest probability score $\mathcal{S}$ respectively. NMS simply removes the duplicated bounding box according to the condition of N_t, but Soft-NMS increases the detection probability of an object by maintaining all the bounding boxes. Fig. 4 compares the algorithms of NMS and Soft-NMS.

In both NMS (2) and Soft-NMS (3), the fixed N_t is used to remove the duplicate detection. A study using N_t was conducted to improve the performance of object detection. Cascade R-CNN [48] confirmed that ample object detection information exists in the case of the $N_t > 0.5$, and the proposed Cascade R-CNN, a multi-step object detection architecture, improved the performance of object detection. Cascade R-CNN completes the three steps of R-CNN for the input image, and object detection is performed by setting the N_t differently for each step. The Tyrolean network (Tnet) [49] is a model used to analyze and learn the prediction box by changing the N_t of each object. Cascade R-CNN and Tnet improved the performance of object detection due to the learning process through several stages but have limitations related to real-time performance. Dual-NMS [50] proposes a dual-NMS structure for object detection in aerial images.

III. PROPOSED ADJUSTABLE-NMS ALGORITHM

Detecting closely overlapped objects in an image is a difficult problem. Therefore, setting N_t to the appropriate value is key to obtaining high accuracy results. In Fig. 5(a) and Fig. 5(b)

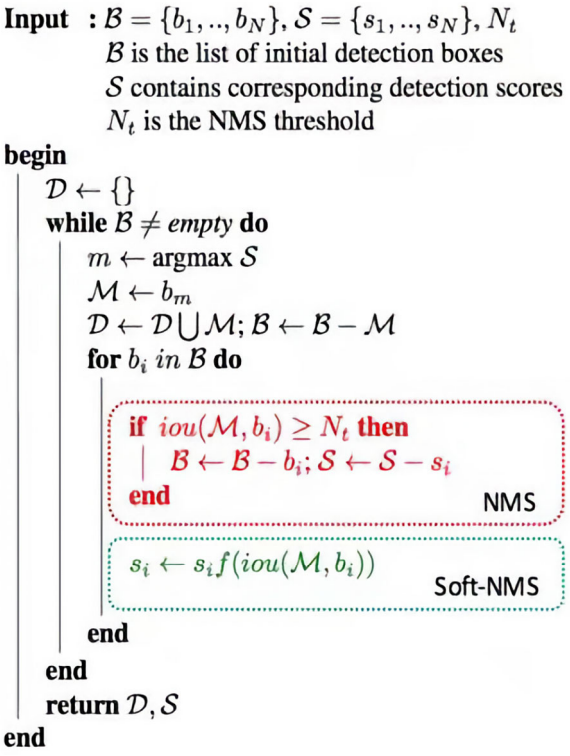


FIGURE 4. Comparing NMS (2) and Soft-NMS (3) algorithms.

N_t is set to 0.50 and 0.59 respectively with varying results. When comparing both results, with N_t=0.50 may remove the positive bounding box on a closely overlapped image or a small image leading to the poor performance.

(a) 7 bicycles detected when setting N_t=0.50

(b) 11 bicycles detected when setting N_t=0.59

FIGURE 5. An example of detecting bicycles in a dense image using different values for N_t. Changing the value of threshold N_t results in more bicycles being detected.

TABLE 1. Comparison of NMS algorithms.

Algorithm	N_t	CS_t	Remarks
NMS	0.5	0.25	Fixed Value
Soft-NMS	0.5	0.25	Fixed Value
Adjustable-NMS	Adjustable, see Section III-B	Adjustable, see Section III-D	Using the object probability score of the input image, different thresholds are used for each image and each object.

Here, we present an adjustable version of the Soft-NMS algorithm (which we call adjustable-NMS) to find the optimum Intersection over Union (IoU) and Confidence Score (CS) thresholds that can minimize the false removal of positive bounding. The performance was compared with the Soft-NMS algorithm. The adjustable-NMS algorithm is based on YOLOv4 and applies the IoU threshold N_t and CS threshold CS_t adjustablely, not fixed, but calculated based on the input image. The proposed adjustable-NMS algorithm with NMS and Soft-NMS were summarized in Table 1.

A. A BRIEF SUMMARY OF PROPOSED ADJUSTABLE NMS ALGORITHM

The Adjustable-NMS algorithm follows a systematic approach to enhance object detection accuracy. The steps involved are as follows:

1) **Calculate Average Probability Score,** $\bar{\mathcal{S}}$: Compute the average probability score based on the individual probability scores, $\mathcal{S}$, of bounding boxes in real time.
2) **Determine IoU Threshold** (N_t): Establish the Intersection over Union (IoU) threshold, N_t in real time.
3) **Update Confidence Scores**: Adjust the probability scores, $\mathcal{S}$, of bounding boxes, in real time, which will be their respective confidence scores (CS). This update ensures that the confidence levels are appropriately reflected.
4) **Calculate Confidence Score Threshold,** CS_t: Determine the threshold CS_t to filter out bounding boxes based on their updated scores in real time.
5) **Identify Objects**: Identify objects in real time where the confidence score exceeds the calculated confidence score threshold CS_t. This step ensures that only the reliable detections are retained.

This method allows for dynamic adjustment of thresholds based on the characteristics of the detected objects, leading to more accurate and reliable object detection, particularly in challenging scenarios with significant overlap. Currently, the scaling factor is fixed, but there is potential to make it adaptive in future iterations, which could further refine detection performance by adjusting more flexibly to varying object densities and distributions.

B. ADJUSTABLE IOU THRESHOLD CALCULATION

The extracted probability score of the bounding box expresses the possibility that an object exists. A high score indicates a high probability of an object in the bounding box. On the

other hand, if the score is low, an object's probability of existing in the bounding box is low. Eq. (4) is an expression of calculating the probability score average $\bar{\mathcal{S}}$ of the extracted bounding boxes. $\mathcal{S}$ represents the probability score of each bounding box, and n represents the number of extracted bounding boxes. Once $\bar{\mathcal{S}}$ is calculated it is used as N_t according to the conditions:

$$\bar{\mathcal{S}} = \sum_{i=1}^{n} (s_i) \div n \tag{4}$$

$$N_t = \begin{cases} 0.8 \cdot \bar{\mathcal{S}}, & \text{if } \bar{\mathcal{S}} \geq 0.5 \\ 1.5 \cdot \bar{\mathcal{S}}, & \text{otherwise} \end{cases} . \tag{5}$$

C. ADJUSTABLE BOUNDING BOX CALCULATION

The bounding box is sorted in descending order based on the probability score. Starting with the bounding box with the highest probability score, the $iou(\cdot)$ of the sorted bounding box is calculated with respect to the adjacent bounding box. Function $iou(\cdot)$ is redefined as follows:

$$iou(A, B) = \frac{A_{area} \cap B_{area}}{A_{area} \cup B_{area}} \tag{6}$$

where A represents a bounding box closer to the ground truth, and B represents an adjacent bounding box of A. The probability score s_i of the B bounding box is updated according to the conditions of the $iou(A, B)$ and N_t. The probability scores are then updated using:

$$s_i = \begin{cases} s_i, & \text{if } iou(A, B) < N_t \\ s_i \cdot (1 - iou(A, B)), & \text{otherwise} \end{cases} \tag{7}$$

and become our Confidence Scores CS.

D. ADJUSTABLE CS THRESHOLD CALCULATION

CS represents the Confidence Score of each bounding box, and n represents the number of extracted bounding boxes. The calculation of our Confidence Score threshold CS_t is relatively straightforward as:

$$CS_t = 2 \left(\frac{\sum_{i=1}^{n} CS}{n} \right) . \tag{8}$$

E. OBJECT DETECTION

If the confidence score CS corresponding to each bounding box is greater than the threshold CS_t, the bounding box is recognized as an object and retained. If the confidence score is less than the threshold, the bounding box is considered overlapped and deleted. The remaining bounding boxes are displayed as object boxes.

IV. RESULTS AND PERFORMANCE ANALYSIS

In this study, Tiny YOLOv4 is used for the performance evaluation of our adjustable-NMS algorithm. Adjustable-NMS performance is compared against the current state-of-the-art Soft-NMS algorithm. All experiments were performed in a Python 3.7 environment using Tensorflow 2.4 on a 2.6GHz CPU computer using Window 10 Pro and a GTX1060Ti GPU graphics card.

The computational complexity of our adjustable-NMS algorithm has no discernible difference in time and space. This is due to the fact that we only utilize the already extracted data from the YOLO training process. There is no difference in memory usage (i.e., space) because we only use the data already extracted during the YOLO training process.

We present two experiments here. The first experiment uses *countable* input images (i.e., have ground truth). This experiment provides evidence that adjustable-NMS performs the same but with better parameter selection than NMS and Soft-NMS. The second experiment shows how adjustable-NMS can detect more objects than Soft-NMS in *dense* input images. During these experiments, we only look at the case of single-class detection. *Countable* indicates that the object density is somewhat sparse enough to allow the number of objects to be easily confirmed by naked eyes. Whereas, *dense* indicates an input image with where object density is so great that it is difficult to visually check the number of objects.

TABLE 2. Thresholds & parameters used in evaluation on *Countable* images.

Image Test	NMS		Soft-NMS		Adjustable-NMS		
	N_t	CS_t	N_t	CS_t	$\bar{\mathcal{S}}$	N_t	CS_t
C1 (in Fig. 6)	0.05	0.25	0.05	0.25	0.70	0.56	0.26
C2 (in Fig. 6)	0.05	0.25	0.05	0.25	0.54	0.43	0.18
C3 (in Fig. 6)	0.05	0.25	0.05	0.25	0.57	0.45	0.18
C4 (in Fig. 6)	0.05	0.25	0.05	0.25	0.58	0.46	0.17
C5 (in Fig. 6)	0.05	0.25	0.05	0.25	0.60	0.48	0.24
C6 (in Fig. 6)	0.05	0.25	0.05	0.25	0.67	0.54	0.20

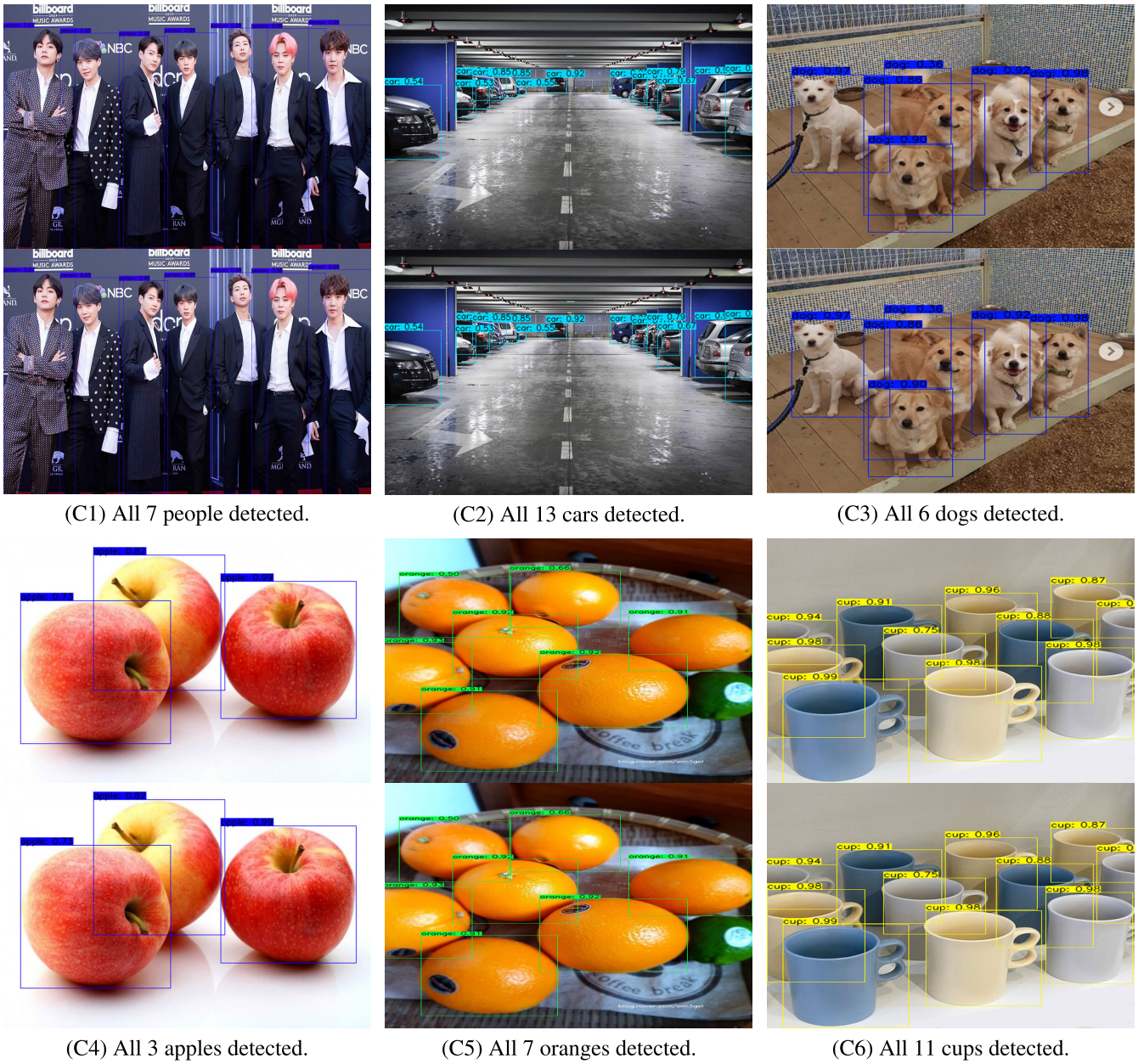
A. COUNTABLE IMAGES EXPERIMENT

Countable input images are images where the object density is somewhat sparse enough to allow the number of objects to be easily confirmed by naked eyes. In this first experiment uses *countable* input images to provide evidence that adjustable-NMS performs the same but with better parameter selection than NMS and Soft-NMS. Having have ground truth, we can visual compare the number of object detected.

We tested both the NMS and Soft-NMS *vs*. Adjustable-NMS Fig. 6 visually shows the detection results and Table 2 show the final threshold and parameter values used for each algorithm. During this experiment, we only look at the case of single-class detection.

It can be seen that $\bar{\mathcal{S}}$ for all the countable input images has a value higher than 0.5. This means that an object is more likely to exist and that it can be easily identified. Note that while the N_t and CS_t thresholds are fixed for both NMS and Soft-NMS algorithms, while the adjustable-NMS algorithm calculated the thresholds in real-time.

The test in Fig. 6 confirmed that adjustable-NMS exhibits the same performance as Soft-NMS in images with somewhat sparse object density. Therefore, it can be concluded that



(C1) All 7 people detected. (C2) All 13 cars detected. (C3) All 6 dogs detected.

(C4) All 3 apples detected. (C5) All 7 oranges detected. (C6) All 11 cups detected.

FIGURE 6. In each sub-figure (*countable* image test), the top image is the result for both NMS and Soft-NMS, and the bottom image is the result for adjustable-NMS. It can be observed that adjustable-NMS performs just as well as NMS and Soft-NMS. However, adjustable-NMS used different threshold values (see Table 2).

adjustable-NMS performs similarly to Soft-NMS; in addition, it can calculate the optimal N_t and CS_t thresholds in real-time.

Our results show that for countable and sparse objects, with published datasets readily available such as COCO and ImageNet [51], [52], there is little discernible difference in performance among traditional NMS, Soft NMS, and adjustable NMS methods across diverse object categories, sizes, and scene types. Table 2 and Figure 6 represent only a subset of the testing images, yet our findings are consistent across a broader range of countable and sparse object images.

B. DENSE IMAGES EXPERIMENT

Dense input images are images where the object density is difficult to visually check; i.e., the total number of objects is unknown. In this second experiment uses *dense* input images to shows how adjustable-NMS can detect more objects than Soft-NMS in these type of images.

Fig. 7 visually shows the detection results of adjustable-NMS and Soft-NMS on nine *dense* input images (D1–D9). A numeric tabulation is given in Table 3. During this experiment, we only look at the case of single-class detection. We consider this experiment to be a difficult object detection

(D1) 13 *vs*. 15 people detected. (D2) 17 *vs*. 22 people detected. (D3) 8 *vs*. 13 people detected.

(D4) 21 *vs*. 25 people detected. (D5) 30 *vs*. 37 people detected. (D6) 7 *vs*. 11 bicycles detected.

(D7) 13 *vs*. 16 apples detected. (D8) 12 *vs*. 18 oranges detected. (D9) 10 *vs*. 12 cups detected.

FIGURE 7. In each sub-figure (*dense* image test), the top image is the result for Soft-NMS, and the bottom image is the result for adjustable-NMS. In every test, adjustable-NMS performs better than Soft-NMS by detecting more of the same type of object (numerical tabulation in Table 3).

TABLE 3. Performance evaluation on *Dense* images.

Test Image	Objects	Soft-NMS			Adjustable-NMS				Difference	Improved By
		N_t	CT_t	Found	$\bar{S}$	N_t	CT_t	Found		
D1 (in Fig. 7)	People	0.05	0.25	13	0.37	0.56	0.17	15	+2	15.4%
D2 (in Fig. 7)	People	0.05	0.25	17	0.43	0.65	0.17	22	+5	29.4%
D3 (in Fig. 7)	People	0.05	0.25	8	0.21	0.31	0.08	13	+5	62.5%
D4 (in Fig. 7)	People	0.05	0.25	21	0.40	0.61	0.20	25	+4	19.0%
D5 (in Fig. 7)	People	0.05	0.25	30	0.40	0.61	0.18	37	+7	23.3%
D6 (in Fig. 7)	Bicycles	0.05	0.25	7	0.40	0.60	0.15	11	+4	57.1%
D7 (in Fig. 7)	Apples	0.05	0.25	13	0.39	0.58	0.16	16	+3	23.1%
D8 (in Fig. 7)	Oranges	0.05	0.25	12	0.30	0.45	0.20	18	+6	50.0%
D9 (in Fig. 7)	Cups	0.05	0.25	10	0.39	0.58	0.13	12	+2	20.0%
Mean Performance Improvement Using adjustable-NMS:									**+4**	**33.3%**(95% CI: 31.5% to 35.1%)

test because the number of similar types of objects (i.e., people) are very densely pack and the true total number of objects in such images is unknown.

The input image we chose where very different in the objects that needed to be detected; for instance, used images people, bicycles, fruits, and cups. Each image has a high object density, making it difficult to visually identify the number of objects that existed in each image. It can be seen that $\bar{S}$ of each input image has a value lower than 0.5 meaning there is a high density of objects with many overlapping objects. The adjustable-NMS algorithm calculates the threshold value in real-time using $\bar{S}$. In Fig. 7 and Table 3, it is confirmed that adjustable-NMS can detect more objects in all nine high-density input image (D1–D9) tests. Adjustable-NMS detects an average of four more objects than Soft-NMS and shows improved performance of more than 33.3% (95% Confidence Interval (CI): 31.5% to 35.1%). This clearly shows that adjustable-NMS can perform much better than Soft-NMS when detecting single-class objects in *dense* input images. However, please note that for dense and overlapped scenarios, we faced challenges in obtaining a sufficiently diverse set of testing images. We acknowledge this limitation and are committed to addressing it. To this end, we plan to expand our evaluation in future work to include a wider variety of object densities and scene types.

Overfitting is not a concern in our approach because the adjustable Non-Maximum Suppression (NMS) algorithm functions as a post-processing step rather than being learning-based. The object classification is handled by a pre-trained YOLO model, which efficiently manages the classification tasks. Overfitting is addressed during the training, validation, and testing phases of YOLO. In the adjustable-NMS algorithm, a key step to enhance the detection of objects of various sizes and in diverse scenes is the real-time calculation and continuous updating of the average probability score for each scene. This adaptive thresholding ensures that the NMS process remains responsive to the unique characteristics of input images, thereby improving the generalizability of the detection system across a wide range of object distributions, densities, and scene types.

While the adjustable-NMS algorithm demonstrates promising results compared to traditional NMS and Soft-NMS, it has potential limitations that need verification. One such limitation is the additional computational overhead from real-time threshold calculations. Although generally minimal and manageable, this process involves performing two additional multiplications for every scene processed, which could impact performance in environments with severely limited computational resources or when handling a high volume of images, such as in real-time applications or on edge computing devices. Additionally, the adjustable-NMS algorithm has not yet been thoroughly tested for multi-class detection. The scaling factor must be adjusted to handle various dense and overlapping scenarios involving multiple objects. Developing strategies to address these complexities will be essential for improving the algorithm's performance in more diverse scenarios, including refining the algorithm to effectively manage different classes and ensuring robust performance across multiple object types within a single image.

Future work will focus on addressing these potential limitations by conducting more extensive experiments to evaluate computational cost, and extending the algorithm's capabilities to support multi-class detection. By tackling these challenges, we aim to enhance the algorithm's robustness and versatility, ultimately improving its effectiveness and applicability in real-world applications.

V. CONCLUSION

We presented the adjustable-NMS algorithm, which employs adjustable IoU and Confidence Score thresholds to enhance object detection in densely populated images. Our innovative approach dynamically adjusts these thresholds based on the density and distribution of objects within the input image, allowing for more accurate detections even in challenging scenarios with significant overlap. The performance evaluation demonstrates a substantial improvement, with the adjustable-NMS algorithm achieving an average 33.3% increase in detection accuracy over the traditional Soft-NMS method.

This significant performance boost underscores the potential of adjustable thresholding in object detection tasks, especially in complex environments where objects are densely packed. By effectively managing overlapping detections, the adjustable-NMS algorithm ensures more reliable

identification and localization of objects, making it a valuable tool for applications requiring high precision.

Looking forward, the algorithm's future enhancements include optimization for real-time systems and improving multi-class object detection, where its dynamic threshold adjustment could significantly boost the efficiency and accuracy of live object detection tasks. Integrating adjustable-NMS with other deep learning components, such as feature extractors or classifiers, is also planned. This integration could enhance performance in high-precision and high-speed tasks, such as autonomous driving, surveillance, and robotics. By leveraging advanced optimization and machine learning techniques, we aim to further automate threshold adjustments, enhancing the algorithm's adaptability and overall effectiveness.

REFERENCES

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[2] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci, "3D object detection from images for autonomous driving: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3537–3556, May 2024.

[3] F. Liu, D. Chen, F. Wang, Z. Li, and F. Xu, "Deep learning based single sample face recognition: A survey," *Artif. Intell. Rev.*, vol. 56, no. 3, pp. 2723–2748, Mar. 2023.

[4] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020.

[5] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A simple and strong anchor-free object detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1922–1933, Apr. 2022.

[6] M. Islam, G. Chen, and S. Jin, "An overview of neural network," *Amer. J. Neural Netw. Appl.*, vol. 5, no. 1, pp. 7–11, 2019.

[7] A. Alsajri, "A review on machine learning strategies for real-world engineering applications," *Babylonian J. Mach. Learn.*, vol. 2023, pp. 1–6, Jan. 2023.

[8] M. M. Mijwel, A. Esen, and A. Shamil, "Overview of neural networks," *Babylonian J. Mach. Learn.*, vol. 2023, pp. 42–45, Aug. 2023.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

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

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[13] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 850–855.

[14] B. Ma, Z. Liu, F. Jiang, Y. Yan, J. Yuan, and S. Bu, "Vehicle detection in aerial images using rotation-invariant cascaded forest," *IEEE Access*, vol. 7, pp. 59613–59623, 2019.

[15] P. Jackson and B. Obara, "Avoiding over-detection: Towards combined object detection and counting," in *Proc. 16th Int. Conf., Artif. Intell. Soft Comput. (ICAISC)*, Zakopane, Poland, Jun. 2017, pp. 75–85.

[16] N. Kim, D. Lee, and S. Oh, "Learning instance-aware object detection using determinantal point processes," *Comput. Vis. Image Understand.*, vol. 201, Dec. 2020, Art. no. 103061.

[17] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6469–6477.

[18] D. Rukhovich, K. Sofiiuk, D. Galeev, O. Barinova, and A. Konushin, "IterDet: Iterative scheme for object detection in crowded environments," in *Proc. IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*, 2021, pp. 344–354.

[19] N. Gählert, N. Hanselmann, U. Franke, and J. Denzler, "Visibility guided NMS: Efficient boosting of amodal object detection in crowded traffic scenes," 2020, *arXiv:2006.08547*.

[20] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023.

[21] Y. Xue, Y. Zhang, Y. Liu, and X. Qian, "Overlapping object detection with adaptive Gaussian sample division and asymmetric weighted loss," *Knowl.-Based Syst.*, vol. 293, Jun. 2024, Art. no. 111685.

[22] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.

[23] H. Zhou, Z. Li, C. Ning, and J. Tang, "CAD: Scale invariant framework for real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 760–768.

[24] S. Liu, D. Huang, and Y. Wang, "Adaptive NMS: Refining pedestrian detection in a crowd," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6452–6461.

[25] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–799.

[26] Y. Huang, Z. Tang, D. Chen, K. Su, and C. Chen, "Batching soft IoU for training semantic segmentation networks," *IEEE Signal Process. Lett.*, vol. 27, pp. 66–70, 2020.

[27] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3588–3597.

[28] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI*, 2020, pp. 12993–13000.

[29] Y. Liu, L. Liu, H. Rezatofighi, T.-T. Do, Q. Shi, and I. Reid, "Learning pairwise relationship for multi-object detection in crowded scenes," 2019, *arXiv:1901.03796*.

[30] Y. Song, Q.-K. Pan, L. Gao, and B. Zhang, "Improved non-maximum suppression for object detection using harmony search algorithm," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105478.

[31] J. Yan, H. Wang, M. Yan, W. Diao, X. Sun, and H. Li, "IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery," *Remote Sens.*, vol. 11, no. 3, p. 286, Feb. 2019.

[32] D. Hema and S. Kannan, "Estimating maximum likelihood using the combined linear and nonlinear function in NMS for object detection," in *Proc. 3rd Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2020, pp. 940–944.

[33] X. Zhang, Z. Yang, F. Shi, Y. Yang, and M. Zhao, "Infrared small target detection based on singularity analysis and constrained random walker," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2050–2064, 2023.

[34] Z. Li, C. Pang, C. Dong, and X. Zeng, "R-YOLOv5: A lightweight rotational object detection algorithm for real-time detection of vehicles in dense scenes," *IEEE Access*, vol. 11, pp. 61546–61559, 2023.

[35] N. Ravi and M. El-Sharkawy, "Addressing the gaps of IoU loss in 3D object detection with IIoU," *Future Internet*, vol. 15, no. 12, p. 399, Dec. 2023.

[36] K. Su, L. Cao, B. Zhao, N. Li, D. Wu, and X. Han, "N-IoU: Better IoU-based bounding box regression loss for object detection," *Neural Comput. Appl.*, vol. 36, no. 6, pp. 3049–3063, Feb. 2024.

[37] J.-Y. Choi and J.-M. Han, "Deep learning (fast R-CNN)-based evaluation of rail surface defects," *Appl. Sci.*, vol. 14, no. 5, p. 1874, Feb. 2024.

[38] 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.*, 2015, pp. 91–99.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[40] J. Fan, J. Lee, I. Jung, and Y. Lee, "Improvement of object detection based on faster R-CNN and YOLO," in *Proc. 36th Int. Tech. Conf. Circuits/Syst., Comput. Commun. (ITC-CSCC)*, Jun. 2021, pp. 1–4.

[41] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[42] A. Farhadi and J. Redmon, "YOLOV3: An incremental improvement," in *Proc. Comput. Vis. Pattern Recognit.*, vol. 1804. Heidelberg, Germany: Springer, 2018, pp. 1–6.

[43] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[44] G. Jocher, "Ultralytics/YOLOv5: V7.0 - YOLOv5 SOTA realtime instance segmentation," *Zenodo*, Nov. 2022, doi: 10.5281/zenodo.7347926.
[45] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
[46] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.
[47] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *Ultralytics YOLOv8*. [Online]. Available: https://github.com/ultralytics/ultralytics
[48] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.
[49] J. Hosang, R. Benenson, and B. Schiele, "A convnet for non-maximum suppression," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 9796, B. Rosenhahn and B. Andres, Ed., Cham, Switzerland: Springer, 2016, pp. 1–14.
[50] Z. Lin, Q. Wu, S. Fu, S. Wang, Z. Zhang, and Y. Kong, "Dual-NMS: A method for autonomously removing false detection boxes from aerial image object detection results," *Sensors*, vol. 19, no. 21, p. 4691, Oct. 2019.
[51] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.
[52] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

KYEONGMI NOH received the M.S. degree in computer science from Korea National Open University, in 2015. She is currently pursuing the Ph.D. degree with the Department of Information and Communication Media Engineering, Seoul National University of Science and Technology. She works as a Firmware Developer at PSK (Semiconductor Equipment Company), South Korea. Her research interests include wireless sensor communication, deep learning, and AI.

SEUL KI HONG received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009, 2011, and 2015, respectively. He was a Senior Engineer with Samsung Electronics, South Korea. He is currently an Assistant Professor with the Department of Semiconductor Engineering, Seoul National University of Science and Technology, Seoul, South Korea.

STEPHEN MAKONIN (Senior Member, IEEE) received the Ph.D. degree in computing science from Simon Fraser University (SFU), Burnaby, Canada, in 2014. He is currently an Adjunct Professor in engineering science, the Principal Investigator of the Computational Sustainability Laboratory, and the Head Instructor in Big Data Hub at SFU. He is a registered Professional Engineer (P.Eng.) with Engineers and Geoscientists BC and has been a Software Engineer for over 24 years working for various local/international industry clients. He is an expert in data science, software engineering, and machine learning. His research interests include computational sustainability and the understanding of socioeconomic issues that pertain to technological advancement. He sits on the IEEE DataPort Advisory Committee and serves as the Editor-in-Chief for the IEEE DataPort Metadata Review Board and an Editorial Board Member for *Scientific Data* (Nature). In addition, he became a Voting Member of the Big Data Governance and Metadata Management (BDGMM, P2957), a new standard being developed by the IEEE Standards Association (IEEE SA) and NIST.

YONGKEUN LEE (Senior Member, IEEE) received the B.S. degree in material engineering from Iowa State University, Ames, IA, USA, in 1991, the M.S. degree in material science from Columbia University, New York, NY, USA, in 1993, and the Ph.D. degree in materials engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1996. His professional journey includes roles, such as an Assistant Professor with Nanyang Technological University, Singapore, and a Principal Engineer with Samsung Electronics LCD Business, South Korea. From 2007 to 2009, he was with the Dean of the Graduate School of Nano IT Design Fusion, Seoul National University of Science and Technology, Seoul, South Korea. He is currently a Professor with the Department of Semiconductor Engineering, Seoul National University of Science and Technology.