Knowledge-Based Systems 241 (2022) 108213

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Joint-attention feature fusion network and dual-adaptive NMS for object detection

Wentao Ma [a], Tongqing Zhou [a], Jiaohua Qin [b,*], Qingyang Zhou [b], Zhiping Cai [a,*]

[a] College of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China
[b] College of Computer Science and Information Technology, Central South University of Forestry & Technology, Changsha, Hunan, 410004, China

ARTICLE INFO

Article history:
Received 27 August 2021
Received in revised form 14 December 2021
Accepted 10 January 2022
Available online 17 January 2022

Keywords:
Object detection
Joint-attention
Adaptive NMS

ABSTRACT

Attention mechanisms and Non-Maximum Suppression (NMS) have proven to be effective components in object detection. However, feature fusion of different scales and layers based on a single attention mechanism cannot always yield gratifying performance, and may introduce redundant information that makes the results worse than expected. NMS methods, on the other hand, generally face the single-constant threshold dilemma, namely, a lower threshold leads to the miss of highly overlapped instance objects while a higher one brings in more false positives. Therefore, how to optimize different dimensions of correlation in feature mapping and how to adaptively set the NMS threshold still hinder effective object detection. While independently addressing each will cause suboptimal detection, this paper proposes to feed the informative feature representation from a joint-attention feature fusion network into adaptive NMS for a comprehensive performance enhancement. Specifically, we embed two types of attention modules in a three-level Feature Pyramid Network (FPN): the channel-attention module is adopted for enhanced feature representation by re-evaluating relationships between channels from a global perspective; the position-attention module is used to exploit the correlation between features to discover rich contextual feature information. Furthermore, we develop dual-adaptive NMS to dynamically adjust the suppression thresholds according to instance objects density, namely, the threshold rises as instance objects gather and decays when objects appear sparsely. The proposed method is evaluated on the COCO dataset and extensive experimental results demonstrate its superior performance compared with existing methods.

1. Introduction

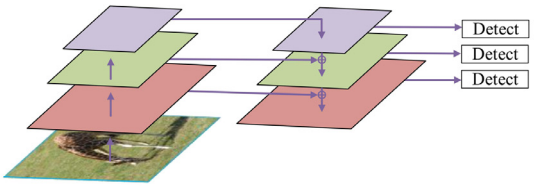
Object detection has been widely studied in computer vision tasks, which heavily rely on the bounding boxes of object category and classification confidence. The development of object detection based on Convolutional Neural Networks (CNN) is comprehensively described in Refs. [1,2]. The most influential CNN-based methods mainly include YOLO [3–7], R-CNN [8–11], SSD [12] and FPN [13,14]. Although these methods have shown gratifying performance, a burning challenging problem of object detection remains the detection of small scale instance objects and dense instance objects.

In a more realistic situation, instance objects in an image often exhibit drastic scale variations. To address this problem, in recent years, multi-level feature fusion techniques, including low-level feature splicing fusion module [12,14,15], multi-scale semantic information fusion module [16–19], attention module [20–29], and feature reuse module [18,30,31] have been widely investigated. Ideally, fusion of multi-level features can compensate for the lack of complementarity between heterogeneous features. However, multiple convolutions on low-level features will inevitably lose some effective information, while the high-level semantic features lack spatial position information, together making the representation ability of features weaker than expected. In particular, for the attention mechanism, using single channel-attention or position-attention alone can hardly make full use of the potential complementary (heterogeneous) information between the multi-level features. Therefore, our approach is motivated to embed a joint-attention module into network architecture. An overview of the three-level feature pyramid network with embedded joint-attention mechanism is shown in Fig. 1.

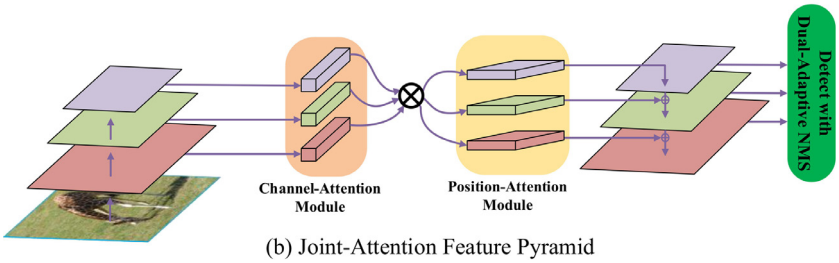
Another concern laid in object detection is the threshold setup of Non-maximum suppression (NMS), which is essentially used to remove redundant detection boxes [32,33]. Generally, the Intersection-over-Union (IoU) is set during the supervision, and all detection boxes of same category are sorted according to their classification confidence scores. Then the box with the highest score is retained, and those neighborhood results

* Corresponding authors.
E-mail addresses: qinjiaohua@163.com (J. Qin), zpcai@nudt.edu.cn (Z. Cai).

https://doi.org/10.1016/j.knosys.2022.108213
0950-7051/



(a) FPN-Style Feature Pyramid



(b) Joint-Attention Feature Pyramid

Fig. 1. Illustration of the joint-attention mechanism embedded in a three-level feature pyramid. The pipeline takes the DarkNet-53 as the backbone and then sequentially adds channel-attention, spatial-attention and self-attention to the three-level feature pyramid.

whose IoU exceeded the threshold are removed. Nevertheless, such a straightforward strategy causes the single constant NMS threshold dilemma: a lower threshold leads to the miss of highly overlapped instance objects, while a higher one brings in more false positives [34]. Many soft NMS variants have been proposed in recent years [33–36]. Instead of discarding all surrounding proposals with IoU below the threshold, they reduce the detection confidence scores of neighborhoods by adding a penalty attenuation function of their overlaps with the higher-scoring bounding box. Although showing promising results in object detection, these soft version NMS methods still inherit the limitations of the single constant NMS threshold.

In light of the above analysis, this paper proposes a novel design for object detection with joint-attention and dual-adaptive NMS as two building blocks. On one hand, the joint-attention module is embedded in a three-level FPN, as shown in Fig. 1. It helps to strengthen the context relation between multi-scale features of the instance objects to enrich the low-level information and high-level semantic information of the fused features. On the other hand, the soft-version NMS is optimized by dynamically setting the NMS threshold in a dual-adaptive manner. The contributions of this work are summarized as follows:

- We propose a joint-attention mechanism for one-stage object detection, which can be expressed as a sequential combination of channel-attention, spatial-attention and aligned self-attention, and embeds these three types of attention into YOLOv3. Such a design renders the network focus on important features and suppress unnecessary ones to make features more informative.
- In object detection task, a constant NMS threshold may eliminate true positives or increase false positives in case of instance objects crowding. We present a dynamic suppression strategy, which adjusts the NMS threshold adaptively according to the aggregation or sparsity of instance objects.
- Extensive experiments are conducted on the COCO dataset with different NMS and attention setups, our method delivers promising improvements in object detection, indicating its effectiveness and superior performance.

The rest of this paper is organized as follows. After a brief review of related work in Section 2, we introduce the joint-attention and dual-adaptive NMS in Section 3. Section 4 presents the experimental results. Finally, conclusions are given in Section 5.

2. Related work

Attention mechanisms and soft version NMS have been proven to be effective in computer vision tasks and have been widely used. In this Section, we will briefly introduce the attention and NMS in object detection.

2.1. Object detection

Object detection based on CNN can be divided into two categories: two-stage detectors with region proposals and one-stage detectors with sliding-window. The former includes two steps for object detection: generation of regional proposals, classification and modification of position, mainly RCNN [8]. This method adopts semantic features extracted by AlexNet, which its performance was improved by nearly 30% compared with the DPM [37] (the highest precision among traditional methods). SPP-Net [13] was proposed based on RCNN, which solves the limitation of input scale and greatly improves the efficiency. Fast RCNN [9] is the unified training of classification loss and boundary box regression loss, so that classification and positioning can share convolution features. However, RCNN, SPP-Net and Fast RCNN still rely on the selective search to extract region proposals, which still have a tremendous computational burden. Faster RCNN [10] realizes the end-to-end training and its detection accuracy and efficiency are greatly improved. Specifically, Mask RCNN [11] and FPN [14] further promote the development of two-stage method.

For low efficiency of two-stage method, one-stage converts object detection into regression problem, among which YOLO [3–5] and SSD [12] are the most classical. Although YOLO has a significant improvement in efficiency, the rigid positioning of the instance objects increases the false positive, leading to less performance than expected. YOLOv2 [4] adopts the pass-through layer and anchor respectively to realize the fusion of multi-level features and improve the positioning. The highly efficient YOLO has a higher accuracy for small objects in an extensive range, named YOLOv3 [5] and YOLOv4 [7], respectively. Inspired by YOLO and Faster RCNN, SSD [12] optimizes the region proposals and feature pyramid network. Although this method hardly considers the potential relationship between different scale features, it dramatically improves the accuracy and efficiency. Generally speaking, both the YOLO series and SSD series one-stage detectors with the leading efficiency and the RCNN series two-stage detectors with the advantage of accuracy have different

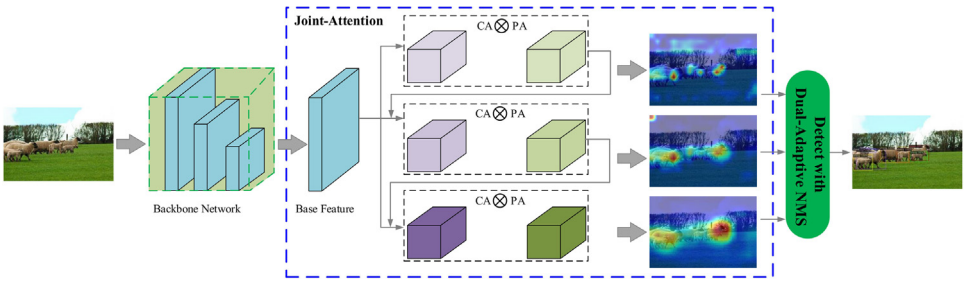


Fig. 2. An overview of the proposed joint-attention feature fusion network and dual-adaptive NMS. Based on FPN-Darknet-53 (an efficient backbone and a feature pyramid network of three levels), we sequentially embed channel-attention, spatial-attention and self-attention to the three-level feature pyramid. Therefore, our method can utilize multiple visual attention mechanisms to perform effective object detection tasks.

degrees of potential class and feature imbalance [1]. Therefore, our method will adopt a joint-attention mechanism based on efficient one-stage method to enhance the channel and position correlation between different scale features, thus improving the performance.

2.2. Attention mechanisms

To the best of our knowledge, the multi-level feature pyramid network feature fusion approach has achieved gratifying achievements in object detection. The main motivation of these methods is to improve feature utilization by obtaining rich spatial details and semantic information, thus improving detection performance for small and dense objects. However, feature fusion also brings about the following two problems. (1) The low-level feature inevitably loses some effective information due to repeated convolution, weakening the final scale feature representation. (2) The fusion feature may be noisy, leading to negative inference for the detector.

To alleviate these challenges, many researchers have introduced the attention mechanism [20–29] into object detection to improve the context relation of multi-scale features to optimize the detection effect. Yi et al. [38] proposed ASSD based on SSD and attention mechanisms. This method introduces the attention module into each scale feature and enhances the attention for important information. In [23], the attention mechanism based on compression and excitation is proposed, which can efficiently process various images in multiple domains without any prior knowledge by the designed adaptive layers. Li et al. [26] proposed a salient object detection based on the multi-attention module, which can guide feature fusion by calculating the weight of scale features. FPA [39] applies the spatial-attention module to high-level semantic features and learns effective feature representations combined with global pool strategies. Moreover, the context feature information obtained by the global attention upsampling module in the decoder layer can guide the lower-level features to filter the location details of the category. Squeeze-and-Excitation Networks (SE-Nets) [22] introduced a lightweight gating mechanism that focuses on enhancing the representational power of the convolutional network by modeling channel-wise relationship. Compared to the SE module, the GCT [29] also pays attention to the cross-channel relationship but can achieve better performance gains with less computation and parameters.

Inspired by these works, this paper designs a joint-attention mechanism including channel-attention, spatial-attention and self-attention, which is embedded into the feature pyramid network of three levels. Channel-attention module and spatial-attention module respectively obtain the relationship between channels and spatial positions, and update specific channel (position) by the weighted sum of all channels (positions). Self-attention is adopted to measure the dependence of visual features between spatial and channel dimensions. Thus, the joint-attention can enrich the semantic and spatial details of multi-scale features and enhance the feature representation of instance objects.

2.3. Non-maximum suppression

NMS is the last but not least important step in most computer vision tasks. It is widely used in feature point detection [40], semantic segmentation [25,39] and object detection [8,10,32,41,42]. However, traditional Greedy-NMS set a hard NMS threshold, which increases false positives. To this end, Bodla et al. [33] proposed a Soft-NMS, which did not directly delete neighborhoods that exceed the hard NMS threshold, but decays their confidence scores by linear weighting or Gaussian weighting. Then the appropriate confidence threshold is selected to remove the bounding box, which greatly reduces the false positives. Specifically, the Softer-NMS [43] averages the selected boxes in a "Softer" manner different from selecting boxes or changing scores, i.e., by calculating the standard deviation between the ground truth locations and predicted locations and combining the neighborhood bounding box via weighted average as the final detection result. The IoU-guided NMS [35] takes the IoU between the predicted bounding box and the ground truth as the location confidence score, and then removes those bounding boxes that are larger than the threshold. Meanwhile, the highest classification confidence score is taken as the final confidence score to preserve the bounding box with more accurate positioning. Liu et al. [34] proposed Adaptive-NMS in the pedestrian detection scene. This method designs a subnetwork that can predict the NMS threshold based on the density of instance objects, which improves the adaptability of the hard threshold to a certain extent.

3. The proposed method

As shown in Figs. 1 and 2, joint-attention and dual-adaptive NMS are proposed by this paper. To implement the joint-attention mechanism, we choose FPN-Darknet-53 as the baseline and integrate the attention modules for object detection. We will further introduce the various parts of the network architecture.

3.1. Backbone network

We take YOLOv3 [5] as the baseline, which is a convolutional network model for object detection with a feature pyramid network of three levels. The convolution feature extracted by YOLOv3 will not weigh the information of each position in the convolution kernel, e.g., it considers that each region contributes equally to the final detection. However, in a realistic situation, there is a lot of complex and rich contextual noise information surrounding the instance objects. Therefore, weighted selection of feature information in the instance objects region can improve the positioning performance of the bounding box.

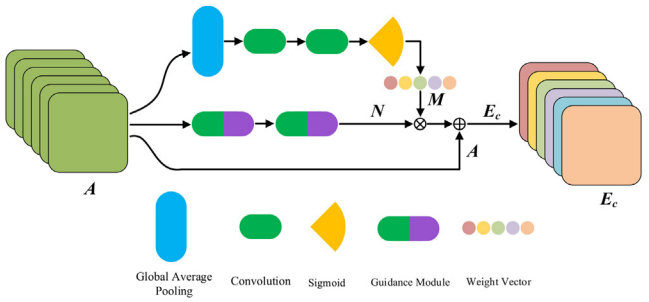


Fig. 3. Details of the channel-attention module. It mainly calibrates the weight of channel importance so that each feature can be enhanced or weakened by the weighting.

3.2. Joint-attention

Based on the above considerations, we modified the FPN-Darknet-53 and designed a joint-attention module that integrates the channel-attention and position-attention into the feature pyramid network. Specifically, inspired by SE-Nets [22] and GCT [29], we add channel-attention to feature pyramid network by adaptive scaling across global and local features, so that gradient information with attention effect covers all scale features. Moreover, referring to the [21,24,26], we added the position-attention to the feature mapping, so as to obtain more contextual feature information and enhance the feature representation.

3.2.1. Channel-attention

By modeling and weighting channel relations in feature mapping, the channel-attention module selects the required features to improve the representational ability of significant features. Firstly, the global average pooling is performed for each level feature map to obtain the global information of each channel. Then, the correlation between the channels is adaptively modeled by two full connection layers and the ReLU and Sigmoid activation functions. Finally, the input channel feature information and the weights of the adaptive learning model are weighted to achieve the weight calibration of the feature response. Hence, with the above structure, channel-attention can selectively focus on important features and suppress unnecessary ones.

The structure of channel-attention module is shown in Fig. 3, given a group of convolutional aggregation pyramid local feature responses $A = [A^1, A^2, A^3, \ldots, A^C]$, where $A_{ij} = [a_{ij}^1, a_{ij}^2, a_{ij}^3, \ldots, a_{ij}^C] \in R^{H_i \times W_j \times C}$ is the largest scale feature mapping at (i, j). And each scale a_{ij} features mapping contain feature information from multiple levels. We adopt global average pooling (squeeze and excitation) to generate channel statistics $Z = [z_1, z_2, z_3, \ldots, z_C] \in R^C$, following the SE module [22]. Meanwhile, we will capture channel dependencies by a sigmoid activation function. Then the parameter of the activation operation is M:

$$M = S(F_{C2} \times R(F_{C1} \times Z)) \tag{1}$$

where S is *Sigmoid* function, R is *ReLU* function, $F_{C1} \in R^{\frac{C}{r} \times C}$ and $F_{C2} \in R^{C \times \frac{C}{r}}$, r is the reduction ratio ($r = 16$ in our experiments). In addition, A generates $N \in R^{C \times H \times W}$ through the guidance module (consisting of two Darknet-53 convolutions), which in turn enables low-features to guide the weighting of high-level features. Then the output is obtained by reweighting the input N with activation parameter M: $\tilde{X}_{ij}^C = M_C \times A_{ij}^C$, where $\tilde{X}_{ij}^C = [\tilde{X}_{ij}^1, \tilde{X}_{ij}^2, \tilde{X}_{ij}^3, \ldots, \tilde{X}_{ij}^C]$. Finally, we apply a softmax layer to obtain the channel-attention mapping E_c:

$$E_c = A + \tilde{X}_{ij} \tag{2}$$

In other words, each feature is enhanced or weakened by the channel weighting.

3.2.2. Position-attention

Generally speaking, the channel-attention module performs weighted on the channel dimension to improve its detection performance. However, in realistic situation, some instance objects are crowded, which will generate false positive. Therefore, inspired by previous work [21,22,44], the position-attention is embedded in the model to compensate for the limitation that the channel-attention unable to obtain the position details, thus enriching the context information and enhancing the feature mapping representation. In this part, we elaborate the position-attention, which consists of spatial-attention and self-attention, as shown in Fig. 4. Both types of attention filter and enhance the feature maps in the spatial dimension, so they are called position-attention.

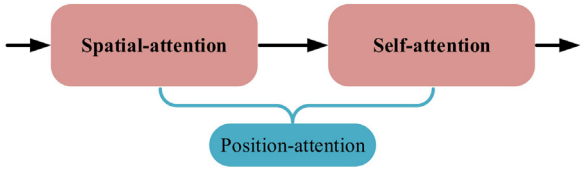


Fig. 4. Details of the position-attention module, which consists of spatial-attention and self-attention.

Spatial-attention. Unlike channel-attention, the spatial-attention module focuses more on "where". Applying pool operations along the channel axis effectively highlights areas of information that complement channel-attention [44]. Firstly, we apply average pooling and max pooling along the channel axis and concatenate them to generate an efficient feature descriptor. Then, the concatenated feature descriptors are encoded by convolution to generate a spatial-attention mapping. Our detailed description is given below.

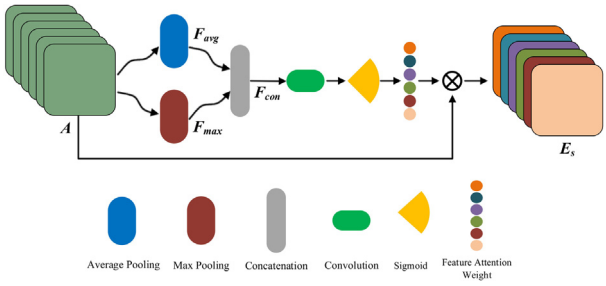


Fig. 5. Details of the spatial-attention module. Different from channel-attention, the spatial-attention module focuses more on "where", which is pooled along the channel axis to highlight effective information area, thus complementing the channel-attention.

The structure of spatial-attention module is shown in Fig. 5, given a group of convolutional aggregation pyramid local feature responses $A = [A^1, A^2, A^3, \ldots, A^C]$, we aggregate the channel information for the feature response by using two pooling operations to generate two mappings. Similarly, F_{avg} and F_{max} are respectively used to represent the output of the two pooling, where global background information and can be selected to highlight salient features of the instance objects. Then the features is concatenated along the channel dimension to obtain F_{con}. Convolution is adopted to reduce dimension to obtain the feature weight, let $C_{3\times3}$ represent convolution with 3×3 kernel size. Finally, we merge the output feature vectors using element-wise

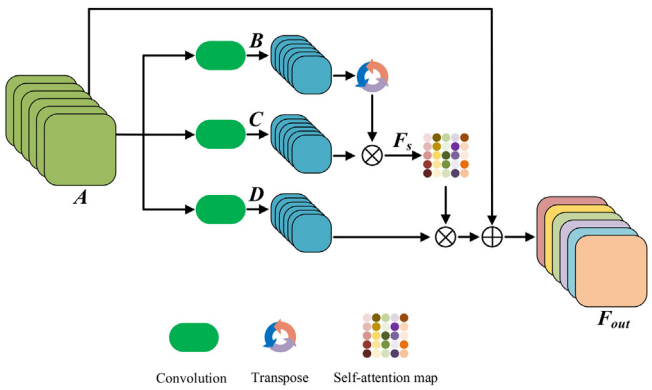


Fig. 6. Details of the self-attention module. The self-attention module can quantify the dependency relationship between any pair of long-range pixels in the feature mapping so as to enrich the context information of instance objects features.

summation. In short, the spatial-attention mapping E_s can be defined as follows:

$$E_s = C_{3\times3}(F_{con}) \times A \tag{3}$$

where $F_{con} \in R^{1\times1\times2C}$ refers to the feature weighting along the channel dimension. In the spatial-attention mapping, the feature in instance objects regions have a high response, and the surrounding information feature are suppressed. As a result, these features become more robust.

Self-attention. Context information is important content to enrich the instance objects features. Therefore, to quantify the dependency relationship between any pair of pixels in the feature mapping, we embed the self-attention into the position-attention module. Inspired by [26,45], the self-attention will calculate the similarity between feature vector and other feature vector in the feature maps, and these similarity scores will generate a weight map with the same dimensions as the input feature. Meanwhile, we multiply the input features by the mapping and sum all the weighted feature vectors to get a new vector, thus updating the original vector.

As shown in Fig. 6, given a convolutional aggregation pyramid local feature $A = [A^1, A^2, A^3, \ldots, A^C] \in R^{C\times H\times W}$, The self-attention module first feeds it to the convolutional layer to generate three features B, C and D, $\{B, C, D\} \in R^{C\times H\times W}$. And we respectively reshape them into $R^{C\times N}$, where $N = H \times W$ represents the number of pixels of the current input feature. Then, the matrix multiplication is performed between transpose of B and C, the softmax function is adopted to calculate the attention space feature map F_s:

$$F_{s_{ji}} = \frac{e^{B_i \cdot C_j}}{\sum_{i=1}^{N} e^{B_i \cdot C_j}} \tag{4}$$

where $F_{s_{ji}}$ measures the ith position's impact on jth position. If more similar feature representations of the two position, which can promote the correlation between them to enrich the information of the feature mapping.

At the same time, we will be performed matrix multiplication between D and the transpose of F_s, and reshape their results for the $R^{C\times H\times W}$. Finally, we multiply the above results by a scale parameter α and perform a element-wise sum with the input feature mapping A to obtain the final output F_{out} as follows:

$$F_{out_j} = \alpha \sum_{i=1}^{N} (F_{s_{ji}} D_i) + A \tag{5}$$

where scale parameter α refers to a variable that is initialized to 0, and it gradually learns to assign optimal weight to different position features in training of the network. According to Eq. (5), the resulting feature F_{out} at each position is a weighted sum of the features by all positions and input features. Therefore, the feature mapping F_{out} has a global receptive field and selectively aggregates contextual information.

3.3. Dual-adaptive NMS

NMS is an important post-processing step based on CNN object detection. In general, Greedy-NMS starts with a set of detection boxes $\mathcal{B}$ with scores $\mathcal{S}$. Then the detection with the maximum score $\mathcal{M}$ is selected and moved from set $\mathcal{B}$ to the set of final detections $\mathcal{D}$. It will also remove any box which has an overlap greater than a threshold N_t with $\mathcal{M}$ in $\mathcal{B}$. This process is repeated for all the remaining boxes in set $\mathcal{B}$. If highly-overlapped, two ground truths can be detected only when setting a large N_t to ensure that the box with the lower confidence score is not suppressed. This is a contradiction: in realistic scenarios, the density of instance objects varies widely, and a higher NMS threshold may increase false positives in regions where instance objects are sparse. To address this issue, many soft NMS variants [32,33,36,43] have been proposed. Rather than discarding all surrounding proposals with scores below the threshold, Soft-NMS [33] reduces neighbor detection scores by adding a penalty function that overlaps with the higher scored bounding box. The re-scoring function of the suppression step on Soft-NMS can be written as:

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i \cdot f(\text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases} \tag{6}$$

where $f(\text{iou}(\mathcal{M}, b_i))$ is an overlapped weighting function, which is adopted to change the classification confidence score s_i of box b_i with a high overlap with $\mathcal{M}$. By Eq. (6), for greedy NMS, $f(\text{iou}(\mathcal{M}, b_i)) \equiv 0$, that is, b_i should be deleted. In Soft-NMS, $f(\text{iou}(\mathcal{M}, b_i))$ $\{f(\text{iou}(\mathcal{M}, b_i)) = (1 - \text{iou}(\mathcal{M}, b_i))$ or $f(\text{iou}(\mathcal{M}, b_i)) = e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}\}$ as a penalty function overlapping with $\mathcal{M}$ to decay the confidence score.

With a soft penalties, if b_i contains another object that is not covered by $\mathcal{M}$, false positive will not be increased at the lower detection threshold. However, as a penalty function, it still allocates a larger penalty for the highly overlapped boxes, similar to the Greedy-NMS penalty. Adaptive-NMS [34] optimizes Soft-NMS for the special crowd scene of pedestrian detection. This method presents a prediction for judging the density of instance objects, which can dynamically increase or decrease the NMS threshold according to the density/sparsity of instance objects. However, although adaptive NMS improves the adaptability of the NMS threshold, the penalty function it adopts is still $f(\text{iou}(\mathcal{M}, b_i)) = (1 - \text{iou}(\mathcal{M}, b_i))$ or $f(\text{iou}(\mathcal{M}, b_i)) = e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}$. For the former, $(1 - \text{iou}(\mathcal{M}, b_i))$ can be tough, especially where the instance objects density is high (the overlapping IoU is greater, but dense). For $e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}$, although it has better decay performance, σ is a variable parameter that needs to be set artificially and lacks adaptability.

According to the above analysis, inspired by Soft-NMS [33] and Adaptive-NMS [34], this paper designs dual-adaptive NMS method, that is, both the decay trend of punishment function and the NMS threshold can be adaptive adjusted. Hence, we define the decay weight of the penalty function as follows:

$$\mathcal{W}_i = \frac{\frac{1}{\text{iou}(\mathcal{M}, b_i)}}{\sum_{i=1}^{k} \frac{1}{\text{iou}(\mathcal{M}, b_i)}} \tag{7}$$

where k represents the number of all boxes that overlap $\mathcal{M}$. It can be seen from Eq. (7) that decay weight of confidence score is positively correlated with iou $(\mathcal{M}, b_i)$, that is, the bounding boxes with small IoU would be hardly affected and the bounding boxes with larger IoU would be assigned a greater penalty. It is the same trend that we set: confidence scores for detection boxes which have a higher overlap with $\mathcal{M}$ should be decayed more, as they have a higher likelihood of being false positives.

With above definition, we propose to update the pruning step with the following strategy:

$$s_i = \begin{cases} s_i, & \text{iou}\,(\mathcal{M}, b_i) < N_{\mathcal{M}} \\ s_i \cdot w_i, & \text{iou}\,(\mathcal{M}, b_i) \geq N_{\mathcal{M}} \end{cases}, \quad \text{where} \sum_{i=1}^{k} w_i = 1 \tag{8}$$

where $N_{\mathcal{M}}$ denotes the adaptive NMS threshold for $\mathcal{M}$, and its adaptive adjustment mechanism is similar to Adaptive-NMS [34], that is, dynamic adjustment is made by the density of the instance objects. Specifically, there are three aspects to note about this strategy. (1) The threshold is also soft, which can adjust the decay adaptively according to the distance between $\mathcal{M}$ and neighboring bounding boxes, so that very close boxes are suppressed to be false positive. It also keeps the correlation between the boxes. In addition, if detection bounding boxes which are far away from $\mathcal{M}$, they are retained the same as the original NMS does, *i.e* $N_{\mathcal{M}} = N_t$. (2) f (iou $(\mathcal{M}, b_i)$) is an overlapping-based weighted penalty function, which has the same computational complexity as Greedy-NMS and Soft-NMS. It is worth noting that the storage of weighted w_i and predicted density have some extra computational overhead, which has little impact on the hardware configuration. (3) Compared with Soft-NMS and Adaptive-NMS, we also improved the hardness of NMS threshold and the awkwardness of σ in the Gaussian penalty function that needs to be manually set to achieve dual-adaptation.

4. Experiments

In this section, we will evaluate our approach on the MS-COCO dataset [46]. We follow a common experimental setting [5,6,24]: the training set, the validation set, and the test-dev set images are 80k, 40k, and 40k, respectively. Specifically, trainval 115k images are adopted for training, and 5k images held out as minival are used for evaluation. For comparison with other state-of-the-art methods, we also showed that mean Average Precision (mAP) over different the NMS thresholds is adopted as the measurements on test-dev split.

This section includes six parts: 4.1. implement details; 4.2. demonstrating the results with channel-attention; 4.3. demonstrating the results with position-attention; 4.4. ablation studies about joint-attention; 4.5. influence of parameter tuning; 4.6. demonstrating the comparisons with state-of-the-art approaches.

4.1. Implementation details

Initialize YOLOv3 (DarkNet-53 [5]) as the backbone networks for feature representations, the entire network is trained with Momentum on 2 GPUs (NVDIA RTX-2080Ti). For experiments based on the attention module, we adopt 5 epochs of warmup strategy for start training. The learning rate is initialized at 1×10^{-3}, and then decrease it to 1×10^{-4} and 1×10^{-5} at 100 epochs and 150 epochs, and stop at 200 epochs. We implement the joint-attention by the Tensorflow framework and the attention modules are added progressively for separate evaluation. Our method is also compared with other state-of-the-art methods based on CNN: one-stage and two-stage. It is worth noting that if not specified the Soft-NMS parameters in [33] are adopted by default, that is, the linear penalty function N_t is set to 0.5, and the Gaussian penalty function σ is set to 0.5

Table 1
Comparison with the performance of different types of NMS methods by embedding multiple channel-attention (including SE module [22], channel-attention of CBAM [21] is represented by CBAM*, GCT [29]).

Methods	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Baseline	33.0	57.9	34.4	18.3	35.4	41.9
SE + Greedy-NMS	33.6	58.4	35.8	18.8	36.7	43.4
CBAM* [21] + Greedy-NMS	33.8	58.6	35.9	18.9	37.2	43.7
GCT [29] + Greedy-NMS	34.0	58.7	36.2	19.1	37.6	43.9
SE + Soft-NMS-L [33]	34.5	59.0	36.8	19.7	37.8	44.5
CBAM* [21] + Soft-NMS-L [33]	34.7	59.1	37.3	20.3	37.9	44.7
GCT [29] + Soft-NMS-L [33]	34.9	59.4	36.8	20.2	38.1	45.1
SE + Soft-NMS-G [33]	34.6	58.9	36.8	20.0	38.0	44.7
CBAM* [21] + Soft-NMS-G [33]	34.8	59.4	37.2	20.2	38.2	45.4
GCT [29] + Soft-NMS-G [33]	35.0	59.5	37.1	20.3	38.5	45.3
SE + Adaptive-NMS [34]	35.0	59.3	37.4	21.2	38.7	45.5
CBAM* [21] + Adaptive-NMS [34]	35.0	59.5	37.5	21.4	38.9	45.6
GCT [29] + Adaptive-NMS [34]	35.4	59.9	37.6	21.5	39.2	45.8
SE + **Our-NMS**	**35.4**	**59.6**	**37.6**	**22.0**	**39.9**	**46.4**
CBAM* [21] + **Our-NMS**	**35.6**	**60.3**	**38.0**	**22.0**	**39.9**	**46.3**
GCT [29] + **Our-NMS**	**35.9**	**60.1**	**37.9**	**22.6**	**40.2**	**46.8**

4.2. Channel-attention

We adopt five kinds of NMS to evaluate the performance of the channel-attention (including SE module [22], channel-attention of FPA [39], channel-attention of CBAM [21], and GCT [29]) mechanism embedded in the model, and the results are shown in Table 1. Baseline is the result of YOLOv3, which uses the Greedy-NMS manner: lower thresholds result in missing highly overlapping instance objects, while higher thresholds result in more false positives. Compared with the baseline, the $AP_{50:95}$ and AP_{50} of SE+Greedy-NMS increased from 33.0% to 33.6% and 57.9% to 58.4%, respectively. The results show that the channel-attention is effective for the object detector without changing NMS.

Moreover, we all know that Greedy-NMS applies a hard threshold to suppress detection boxes, which increases the false positives. Therefore, the Soft-NMS is used instead of Greedy-NMS, which has a certain performance improvement regardless of the linear penalty 'L' or the Gaussian penalty 'G'. Specifically, adaptively adjusting the NMS threshold for SE+Adaptive-NMS also provides some performance improvement. Finally, compared to the baseline, for NMS proposed by this paper, the $AP_{50:95}$ and AP_{50} are improved by 2.4% and 1.7%, respectively. Compared to the SE module, the GCT [29] also pays attention to the cross-channel relationship but can achieve better performance gains with less computation and parameters.

4.3. Position-attention

To verify the effectiveness of the position-attention mechanism, we embedded position-attention consisting of spatial-attention and self-attention in the baseline, as shown in Table 2. Compared with the baseline, the embedded position-attention module increased the $AP_{50:95}$ from 33% to 35.3%, by 2.3 points, without changing the Greedy-NMS. Moreover, we adopt the experimental setting similar to channel-attention and verify the effect with Soft-NMS and Adaptive-NMS respectively, which the performance improvement trend is the same as channel-attention. Specifically, the $AP_{50:95}$ and AP_{50} of "PA+Our-NMS" is 38.2% and 60.7%, respectively. It is worth noting that the $AP_{50:95}$ improvement is 5.2 points compared to the baseline. We believe this is reasonable because the position-attention (spatial-attention and self-attention) allows for richer contextual information to improve feature gaps for small instance objects and to learn the relative importance of relationships between instance objects and context.

Table 2
Comparison with the performance of different types of NMS methods by embedding multiple spatial modules (including spatial-attention of CBAM [21] is represented by CBAM#, FPA [39] and our PA).

Methods	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Baseline	33.0	57.9	34.4	18.3	35.4	41.9
FPA [39] + Greedy-NMS	33.8	58.5	36.1	18.9	36.9	43.6
CBAM# [21] + Greedy-NMS	34.7	59.0	36.2	19.5	37.7	44.8
PA [Ours] + Greedy-NMS	**35.3**	**59.6**	**36.7**	**19.8**	**39.2**	**46.2**
FPA [39] + Soft-NMS-L [33]	34.6	59.0	36.9	19.9	38.0	44.9
CBAM# [21] + Soft-NMS-L [33]	35.6	59.7	37.3	20.0	39.0	46.6
PA [Ours] + Soft-NMS-L [33]	**36.5**	**59.8**	**37.8**	**20.2**	**40.1**	**47.1**
FPA [39] + Soft-NMS-G [33]	34.6	59.4	36.9	20.2	38.3	45.1
CBAM# [21] + Soft-NMS-G [33]	35.7	59.7	37.4	19.6	39.6	46.9
PA [Ours] + Soft-NMS-G [33]	**36.6**	**59.8**	**37.5**	**20.0**	**40.4**	**47.5**
FPA [39] + Adaptive-NMS [34]	35.3	59.5	37.6	21.2	38.7	45.9
CBAM# [21] + Adaptive-NMS [34]	36.4	60.4	38.7	20.6	40.5	47.3
PA [Ours] + Adaptive-NMS [34]	**37.1**	**60.3**	**38.0**	**20.4**	**41.2**	**48.3**
FPA [39] + **Our-NMS**	**35.4**	**59.7**	**37.6**	**22.2**	**40.3**	**46.8**
CBAM# [21] + **Our-NMS**	**37.0**	**61.0**	**38.9**	**21.8**	**41.2**	**48.4**
PA [Ours] + Our-NMS	**38.2**	**60.7**	**38.7**	**21.3**	**41.9**	**49.2**

4.4. Joint-attention

In order to illustrate the effectiveness of each attention module, we conducted the ablation studies on different attention modules respectively, and the results are shown in Tables 3 and 4. The detection accuracy is placed in the order as "Baseline+SA_1, Baseline+SA_2, Baseline+SE, Baseline+CBAM*, Baseline+GCT, Baseline+CBAM, Baseline+SA_1+SA_2, Baseline+SA_1+SE, Baseline+SA_1+CBAM*, Baseline+SA_1+GCT, Baseline+SA_2+SE, Baseline+SA_2+CBAM*, Baseline+SA_2+GCT, Baseline+PA+SE, Baseline+PA+CBAM*, Baseline+PA+GCT and Baseline+SA_2 +CBAM", which all adopt Soft NMS-G manner. Baseline denotes DarkNet-53 backbone, where SA_1 represents spatial-attention, SA_2 denotes self-attention, channel-attention of CBAM is represented by CBAM*. As shown in Table 3, we adopted the "Soft-NMS-G" manner to demonstrate the effects of the various attention modules. SA_1 and SA_2 are integrated with SE and embedded into FPN, which the mAP is improved by 5.6 and 5.4 points, respectively. This indicates that joint attention is effective for detector. Moreover, by embedding the joint-attention composed of SE and PA, the detection accuracy increased from 33% to 40.7%, by 7.7 points. It is worth noting that if the channel-attention GCT is used to replace SE, the detection accuracy increased from 33% to 41.5%, by 8.5 points.

As can be seen from Table 4, with Darknet-53 as the backbone without changing Greedy-NMS, while embedding SA_1 and SA_2, the $AP_{50:95}$ increases to 34.5 and 34.7, respectively. The detection accuracy is increased to 39.7 by both the SE and PA are embedded, 6.7 points higher than the baseline. In addition, we further evaluated the performance of the embedded attention module in Soft-NMS, Adaptive-NMS and Our-NMS. The results show that the embedded attention module can significantly improve the detection accuracy. It is vital to notice that the $AP_{50:95}$ with Our-NMS is 41.4%, which is the state-of-the-art Darknet-53 as backbone object detection accuracy on COCO 5k-validation dataset.

4.5. Parameter tuning

In our experiments, the Soft-NMS parameter in [33] is adopted by default, that is, the NMS threshold $N_t = 0.5$ and the Gaussian penalty function $\sigma = 0.5$. N_t and σ will be fine-tuned in a step size of 0.1 to observe their effects on Soft-NMS, Adaptive-NMS.

As we know, there are two parameters about Soft-NMS: N_t and σ, which control the bouning box overlap thresholds and improve the Gaussian penalty function to locate the instance objects, respectively. Adaptive-NMS can dynamically adjust N_t according to the density of the instance objects, so it is relatively adjustable for one parameter. Fine-tuning parameters by three conditions of Adaptive-NMS (adjust σ), Soft-NMS N_t ($\sigma = 0.5$, adjust N_t) and Soft-NMS σ ($N_t = 0.5$, adjust σ). As shown in Fig. 7, the AP is gradually increased between 0.2 to 0.6 for both detectors, which is the same trend as Soft-NMS [33]. It is worth noting that the Adaptive-NMS performs better than the Soft-NMS (about 1%) irrespective of the value of the selected N_t and σ. We also observe another characteristic that fine-tuning σ has a positive gain. Specifically, at $N_t > 0.5$ and $\sigma > 0.5$, the average precision decreases and is almost stable respectively.

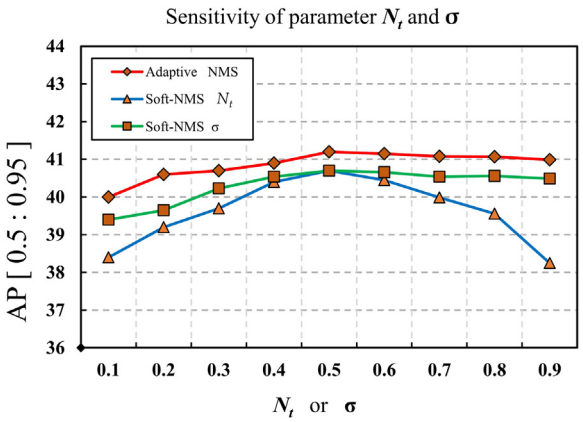


Fig. 7. Sensitivity to σ and N_t on Soft-NMS, Adaptive-NMS.

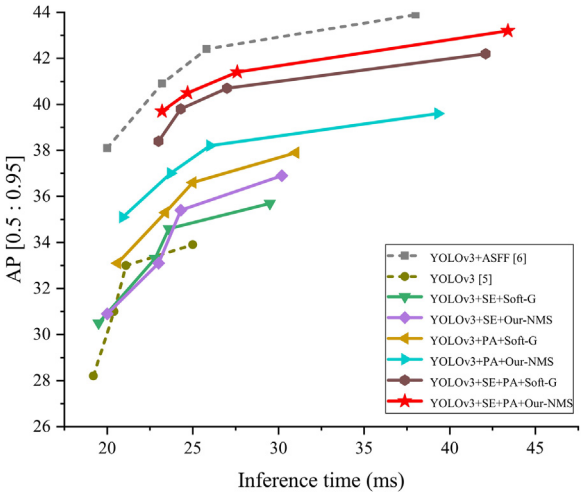


Fig. 8. Speed–accuracy trade-off of the real-time detectors on the MS-COCO test-dev. Instantiated with the one-stage detector YOLOv3, our proposed joint-attention and dual-adaptive NMS outperform some of the state-of-the-art methods.

4.6. Comparison with state-of-the-art

Finally, we compare the experimental results of the joint-attention and dual-adaptive NMS with state-of-the-art object detection methods on MS-COCO test-dev split in Table 5 and Fig. 8. The comparison involves the type of backbone, the input size of the model, and the test results. It is worth noting that the "Join-Attention+Soft-NNS-G" by Darknet-53 achieves $AP_{50:95}$ of 40.7%, which outweighs some object detectors that have deeper backbones and larger input size, e.g., the two-stage method of ResNet-101 as the backbone, Faster RCNN [10] and Mask RCNN [11],

Table 3
Results for Baseline+SA_1, Baseline+SA_2, Baseline+SE, Baseline+CBAM*, Baseline+GCT, Baseline+CBAM, Baseline+SA_1+SA_2, Baseline+SA_1+SE, Baseline+SA_1+CBAM*, Baseline+SA_1+GCT, Baseline+SA_2+SE, Baseline+SA_2+CBAM*, Baseline+SA_2+GCT, Baseline+PA+SE, Baseline+PA+CBAM*, Baseline+PA+GCT and Baseline+SA_2+CBAM, which all adopt Soft NMS-G manner. Baseline denotes DarkNet-53 backbone, SA_1 represents spatial-attention, SA_2 denotes self-attention and channel-attention of CBAM is represented by CBAM*.

Baseline	SA_1	SA_2	SE	CBAM*[21]	GCT [29]	CBAM [21]	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓							33.0	57.9	34.4	18.3	35.4	41.9
✓	✓						35.8	59.2	37.3	19.6	39.6	46.9
✓		✓					35.4	59.4	36.7	19.3	39.7	47.2
✓			✓				34.6	58.9	36.8	20.0	38.0	44.7
✓				✓			34.8	59.4	37.2	20.2	38.2	45.4
✓					✓		35.0	59.5	37.1	20.3	38.5	45.3
✓						✓	38.9	60.5	39.2	25.6	44.0	49.6
✓	✓	✓					36.6	59.8	37.5	20.0	40.4	47.5
✓	✓		✓				38.6	59.7	38.5	23.4	42.5	48.3
✓	✓			✓			39.0	60.3	39.0	23.3	43.0	49.1
✓	✓				✓		39.2	60.4	39.6	24.1	43.5	50.0
✓		✓	✓				38.4	59.9	38.5	23.9	43.7	48.0
✓		✓		✓			39.4	60.2	39.8	23.6	42.9	49.3
✓		✓			✓		39.5	60.7	39.3	23.4	43.9	50.8
✓	✓	✓	✓				40.7	61.2	40.9	24.1	44.3	50.4
✓	✓	✓		✓			40.8	61.5	41.2	24.9	45.0	50.5
✓	✓	✓			✓		41.5	62.3	40.8	25.4	45.8	51.0
✓		✓				✓	41.2	62.0	43.1	25.3	46.1	50.8

Table 4
Ablation study of joint-attention over different NMS manners. The AP is placed in Baseline+SA_1 → Baseline+SA_2 → Baseline+SE+SA_1 → Baseline+SE+SA_2 → Baseline+SE+PA order.

Methods	$AP_{50:95}$	AP_{50}	AP_{75}
Greedy-NMS	34.5 → 34.7 → 36.9 → 37.3 → 39.7	58.8 → 58.6 → 59.4 → 59.7 → 60.3	35.9 → 36.2 → 36.7 → 36.3 → 39.1
Soft-NMS-L	35.7 → 35.7 → 38.4 → 38.2 → 40.3	59.3 → 59.1 → 59.7 → 59.8 → 60.8	37.0 → 36.8 → 38.2 → 38.2 → 40.4
Soft-NMS-G	35.8 → 35.4 → 38.6 → 38.4 → 40.7	59.2 → 59.4 → 59.7 → 59.9 → 61.2	37.3 → 36.7 → 38.5 → 38.5 → 40.9
Adaptive-NMS	36.6 → 36.7 → 39.5 → 39.4 → 41.2	60.0 → 60.2 → 60.6 → 60.3 → 61.5	38.4 → 38.6 → 39.7 → 39.5 → 41.7
Our-NMS	**37.2 → 37.2 → 39.3 → 39.6 → 41.4**	**60.2 → 60.2 → 60.8 → 60.7 → 61.8**	**38.6 → 38.6 → 39.7 → 39.9 → 42.4**

improved its $AP_{50:95}$ by 4 points compared with the former. Since ResNet-101 is the backbone with deeper and larger inputs, the mAP of "Joint-Attention+Soft-NMS-G" is slightly higher than Fitness-NMS [47] and lower than Cascade RCNN [48], but "Joint-Attention+Adaptive-NMS" is slightly higher than the latter. Compared with the one-stage methods such as YOLOv3 [5], LRF [30], PFPnet-R [49] and SSD [12], the detection performance is also greatly improved. Specifically, compared to the original YOLOv3, the mAP of "Joint-Attention+Dual-NMS" improved by nearly 11 points, which shows the effectiveness of our method. Besides, detection accuracy is also comparable to instance objects detection methods of multi-scale strategy such as M2det [18] (with size = 800), YOLOv3+ASFF [6] (with size = 800), and YOLOv4 [7] (with size = 608). Note that the performance gain given by the multi-scale strategy is complementary to the feature fusion and embedding the attention mechanism, which further improves the performance.

For DarkNet-53 backbone, we embed five kinds of NMS (e.g., Greedy-NMS, Soft-NMS-L [33], Soft-NMS-G [33], Adaptive-NMS [34] and Our-NMS) into the YOLOv3 respectively. Our method can achieve superior performance with the same deeper of baseline backbone, reporting 39.7% AP at 40.2 FPS, 40.3% AP at 36.7 FPS, 40.7% AP at 37 FPS, 41.2% AP at 36.5 FPS and 41.4% AP at 36.2 FPS. It is worth noting that compared to baseline YOLOv3, fixing the NMS and embedding SE or PA will increase the inference time of detector. We believe that this is reasonable, because embedding two types of attention modules in a three-level FPN can greatly enrich feature representation, but it will increase the complexity of detector and lead to inference latency.

Furthermore, we also performed the speed–accuracy trade-off for the real-time detectors on the MS-COCO test-dev, as shown in Fig. 8. We adopt the same training model weight parameters to evaluate by different input resolutions. And our method embeds attention mechanisms and dual-adaptive NMS, the efficiency of our method is only slightly lower than that of YOLOv3 [5] and YOLOv3+ASFF [6] when we reduce input image resolution to pursue faster detector. And compared to YOLOv3, with the same inference efficiency our approach improves the performance more significantly. Meanwhile, there is one aspect to note about our dual-adaptive NMS. As shown in Eq. (8), $f(\text{iou}(\mathcal{M}, b_i))$ is an overlapping-based weighted penalty function, which has the same computational complexity as Greedy-NMS and Soft-NMS. It is worth noting that the storage of weighted w_i and predicted density have some extra computational overhead, which has little impact on the hardware configuration. As shown in Table 5, the latency of our dual-adaptive NMS should be compared with other NMS baseline methods, we can observe that the FPS of variant NMS is similar under the same embed attention modules.

5. Conclusions

This paper proposes joint-attention and dual-adaptive NMS instance objects detection. On the one hand, this method can use three attention modules to guide the selection and fusion of features. Specifically, the channel-attention can capture the global dependencies in the channel dimension to guide the fusion of low-level features, while the position-attention can effectively capture long-range dependencies between any pair of pixels to get sufficient contextual information so that similar feature vectors contribute to mutual improvement. On the other hand, for instance objects in scenarios with density differences, our method employs a new dual-adaptive NMS, which can dynamically adjust the NMS threshold according to the density of instance objects. Experiments on COCO dataset demonstrates that the joint-attention and dual-adaptive NMS can achieve superior performance.

Table 5
Comparison of our method with the state-of-the-art object detection method in terms of latency (FPS) and average precision (AP) on *COCO-test-dev*.

	Methods	Backbone	Size	FPS	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods	Faster R-CNN [10]	ResNet-101	800	–	36.7	54.8	39.8	19.2	40.9	51.6
	Mask R-CNN [11]	ResNet-101	640	7.9	38.2	60.3	41.7	20.1	41.1	50.2
	Mask R-CNN [11]	ResNeXt-101	640	6.5	39.8	62.3	43.4	22.1	43.2	51.2
	Fitness-NMS [47]	ResNet-101	1024	5.0	41.8	60.9	44.9	21.5	45.0	57.5
	Cascade R-CNN [48]	Res101-FPN	1280	5.0	42.8	62.1	46.3	23.7	45.5	55.2
One-stage methods	YOLOv3[5]	Darknet-53	608	56	33.0	57.9	34.4	18.3	35.4	41.9
	SE+Greedy-NMS	Darknet-53	608	47.2	33.6	58.4	35.8	18.8	36.7	43.4
	SE+Soft-NMS-L [33]	Darknet-53	608	46.6	34.5	59.0	36.8	19.7	37.8	44.5
	SE+Soft-NMS-G [33]	Darknet-53	608	46.7	34.6	58.9	36.8	20.0	38.0	44.7
	SE+Adaptive-NMS [34]	Darknet-53	608	45.4	35.0	59.3	37.4	21.2	38.7	45.5
	SE+**Our-NMS**	Darknet-53	608	44.1	**35.4**	**59.6**	**37.6**	**22.0**	**39.9**	**46.4**
	YOLOv3+ASFF*[6]	Darknet-53	608	45.5	42.4	63.0	47.4	25.5	45.7	52.3
	YOLOv4[7]	CSPDarknet-53	608	62	43.5	65.7	47.3	26.7	46.7	53.3
	SSD [12]	VGG-16	512	22	28.8	48.5	30.3	10.9	31.8	43.5
	PA+Greedy-NMS	Darknet-53	608	44.3	35.3	59.6	36.7	19.8	39.2	46.2
	PA+Soft-NMS-L [33]	Darknet-53	608	43.6	36.5	59.8	37.8	20.2	40.1	47.1
	PA+Soft-NMS-G [33]	Darknet-53	608	43.7	36.6	59.8	37.5	20.0	40.4	47.5
	PA+Adaptive-NMS [34]	Darknet-53	608	42.0	37.1	60.3	38.0	20.4	41.2	48.3
	PA+**Our-NMS**	Darknet-53	608	41.4	**38.2**	**60.7**	**38.7**	**21.3**	**41.9**	**49.2**
	M2det [18]	VGG-16	800	–	44.2	64.6	49.3	29.2	47.9	55.1
	PFPNet-R [49]	VGG-16	512	–	35.2	57.6	37.9	18.7	38.6	45.9
	RetinaNet [50]	ResNet-101	512	22.3	33.0	54.5	35.5	16.3	36.3	44.3
	RetinaNet [50]	ResNet-101	800	9.3	39.1	59.1	42.3	21.8	42.7	50.2
	EfficientDet-D1[51]	Efficient-B1	640	50.0	39.6	58.6	42.3	17.9	44.3	56.0
	EfficientDet-D2[51]	Efficient-B2	768	41.7	43.0	62.3	46.2	22.5	47.0	58.4
	RetinaMask [52]	ResNet-101-FPN	800	6.9	41.4	60.8	44.6	23.0	44.5	53.5
	Joint-Attention+Greedy-NMS	Darknet-53	608	40.2	39.7	60.3	39.1	23.7	42.4	46.5
	Joint-Attention+Soft-NMS-L [33]	Darknet-53	608	36.7	40.3	60.8	40.4	24.4	44.2	50.1
	Joint-Attention+Soft-NMS-G [33]	Darknet-53	608	36.7	40.7	61.2	40.9	24.1	44.3	50.4
	Joint-Attention+Adaptive-NMS [34]	Darknet-53	608	36.5	41.2	61.5	41.7	24.6	44.2	51.9
	Joint-Attention+Our-NMS	Darknet-53	608	36.2	**41.4**	**61.8**	**42.4**	**25.1**	**44.6**	**53.7**

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61772561, 62002392, 62072465 and 62172155; in part by the Key Research and Development Plan of Hunan Province under Grant 2019SK2022; in part by the Postgraduate Excellent teaching team Project of Hunan Province under Grant [2019]370-133; in part by the Natural Science Foundation of Hunan Province, China under Grant 2020JJ4141 and 2020JJ4140; in part by the Postgraduate Research and Innovation Project of Hunan Province under Grant CX20210080.

References

[1] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: A survey, Int. J. Comput. Vis. 128 (2) (2020) 261–318.
[2] M.E. Villa-Pérez, M.A. Álvarez-Carmona, O. Loyola-González, M.A. Medina-Pérez, J.C. Velazco-Rossell, K.-K.R. Choo, Semi-supervised anomaly detection algorithms: A comparative summary and future research directions, Knowl.-Based Syst. (2021) 106878.
[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[4] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[5] A. Farhadi, J. Redmon, Yolov3: An incremental improvement, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[6] S. Liu, D. Huang, Y. Wang, Learning spatial fusion for single-shot object detection, 2019, arXiv preprint arXiv:1911.09516.
[7] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection, 2020, arXiv preprint arXiv:2004.10934.
[8] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[9] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[11] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[13] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: European Conference on Computer Vision, Springer, 2014, pp. 346–361.
[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[15] X. Wei, S. Liu, Y. Xiang, Z. Duan, C. Zhao, Y. Lu, Incremental learning based multi-domain adaptation for object detection, Knowl.-Based Syst. 210 (2020) 106420.
[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: Deconvolutional single shot detector, 2017, arXiv preprint arXiv:1701.06659.
[17] Z. Li, F. Zhou, Fssd: feature fusion single shot multibox detector, 2017, arXiv preprint arXiv:1712.00960.
[18] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, H. Ling, M2det: A single-shot object detector based on multi-level feature pyramid network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 9259–9266.
[19] L. Yang, C. Kong, X. Chang, S. Zhao, Y. Cao, S. Zhang, Correlation filters with adaptive convolution response fusion for object tracking, Knowl.-Based Syst. 228 (2021) 107314.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 5998–6008.
[21] S. Woo, J. Park, J.-Y. Lee, I. So Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[22] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[23] X. Wang, Z. Cai, D. Gao, N. Vasconcelos, Towards universal object detection by domain attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7289–7298.
[24] Y. Li, S. Wang, HAR-net: Joint learning of hybrid attention for single-stage object detection, 2019, arXiv preprint arXiv:1904.11141.
[25] Z.-L. Ni, G.-B. Bian, G.-A. Wang, X.-H. Zhou, Z.-G. Hou, H.-B. Chen, X.-L. Xie, Pyramid attention aggregation network for semantic segmentation of surgical instruments, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11782–11790.
[26] A. Li, J. Qi, H. Lu, Multi-attention guided feature fusion network for salient object detection, Neurocomputing 411 (2020) 416–427.
[27] R. Chen, Y. Xie, X. Luo, Y. Qu, C. Li, Joint-attention discriminator for accurate super-resolution via adversarial training, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 711–719.
[28] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3146–3154.
[29] Z. Yang, L. Zhu, Y. Wu, Y. Yang, Gated channel transformation for visual recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 11794–11803.
[30] T. Wang, R.M. Anwer, H. Cholakkal, F.S. Khan, Y. Pang, L. Shao, Learning rich features at high-speed for single-shot object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1971–1980.
[31] X. Li, Q. Liu, N. Fan, Z. He, H. Wang, Hierarchical spatial-aware siamese network for thermal infrared object tracking, Knowl.-Based Syst. 166 (2019) 71–81.
[32] J. Hosang, R. Benenson, B. Schiele, Learning non-maximum suppression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4507–4515.
[33] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-nms–improving object detection with one line of code, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5561–5569.
[34] S. Liu, D. Huang, Y. Wang, Adaptive nms: Refining pedestrian detection in a crowd, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6459–6468.
[35] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 784–799.
[36] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2888–2897.
[37] Y. Tang, X. Wang, E. Dellandréa, L. Chen, Weakly supervised learning of deformable part-based models for object detection via region proposals, IEEE Trans. Multimed. 19 (2) (2016) 393–407.
[38] J. Yi, P. Wu, D.N. Metaxas, Assd: Attentive single shot multibox detector, Comput. Vis. Image Underst. 189 (2019) 102827.
[39] H. Li, P. Xiong, J. An, L. Wang, Pyramid attention network for semantic segmentation, 2018, arXiv preprint arXiv:1805.10180.
[40] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. J. Comput. Vis. 60 (1) (2004) 63–86.
[41] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, in: AAAI, 2020, pp. 12993–13000.
[42] R. Rothe, M. Guillaumin, L. Van Gool, Non-maximum suppression for object detection by passing messages between windows, in: Asian Conference on Computer Vision, Springer, 2014, pp. 290–306.
[43] Y. He, X. Zhang, M. Savvides, K. Kitani, Softer-nms: Rethinking bounding box regression for accurate object detection, 2, 2018, arXiv preprint arXiv:1809.08545.
[44] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2016, arXiv preprint arXiv:1612.03928.
[45] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.
[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[47] L. Tychsen-Smith, L. Petersson, Improving object localization with fitness nms and bounded iou loss, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6877–6885.
[48] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.
[49] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, S.-J. Ko, Parallel feature pyramid network for object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 234–250.
[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[51] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[52] C.-Y. Fu, M. Shvets, A.C. Berg, Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free, 2019, arXiv preprint arXiv:1901.03353.