Multi-granularity Detector for Enhanced Small Object Detection under Sample Imbalance

Chen Dong

Tongji University, No.4800, Cao'an Highway, Jiading District, Shanghai
1910691@tongji.edu.cn

Miao Duoqian*

Tongji University, No.4800, Cao'an Highway, Jiading District, Shanghai
dqmiao@tongji.edu.cn

Zhao Xuerong

Shanghai Normal University, No. 100 Haisi Road, Fengxian District, Shanghai
xrzhao@shnu.edu.cn

Abstract

This paper delves into the realm of object detection models, pinpointing challenges posed by inadequate performance in discerning small objects and the inherent imbalance between positive and negative samples. In response, we introduce the Multi-Granularity Detector (MgD), a sophisticated fusion of Multi-Granularity Feature Extraction (MFE) and Sequential Three-Way Selection (S3WS). Within the MFE framework, three multi-granularity customizable deformable convolutions span three layers of feature maps, meticulously tailored for nuanced object analysis across diverse size spectrums. This innovative approach notably enhances small object detection accuracy, cascading improvements to overall object detection efficacy. Simultaneously, the S3WS mechanism is introduced to rectify the imbalance between positive and negative samples. Within this framework, region proposals undergo scrutiny, with additional positive samples judiciously selected from positive and boundary regions. This selection process relies on multiple evaluation functions and two dynamic thresholds, strategically applied layer by layer. Exhaustive experiments on the COCO benchmark unequivocally establish MgD as a superior performer at the system level. Notably, SwinV2-G, enhanced with MFE and SW3S (AP 63.1→64.0, AP/AP_s 1.97→1.42), surpasses prevailing state-of-the-art results. MgD [1] (AP 53.9, AP/AP_s 1.35) significantly enhances the detection of small objects. Additionally, MFE and S3WS can be seamlessly integrated into ConvNet detectors and transformer-based detectors, achieving significant improvements.

Keywords: Computer Vision, Deep Learning, Object Detection, Granular Computing, Three-way Decisions

1. Introduction

Object detection endeavors to identify objects within an image, discerning their respective classes and spatial coordinates. Convolutional neural networks (ConvNets [1]) have been instrumental in propelling advancements in object detection. The adoption of progressively deeper neural networks and intricate convolutional structures has notably elevated detection

*Corresponding author
[1] Implementation codes are publicly available at https://github.com/Alan-D-Chen/MgD

Table 1: Detection results (%) on MS COCO *test-dev* set. AP denote the average precision of all categories, AP_s for small objects, AP_m for medium objects and AP_l for large objects. AP/AP_s represents the gap between AP and AP_s. The closer AP/AP_s(proportion) is to one, the greater the contribution of AP_s. Table 1 displays that AP_s severely restrict AP and representative models ignore this problem.

Method	AP	AP_s	AP_m	AP_l	AP/AP_s
anchor-based two-stage					
MLKP	28.6	**10.8**	33.4	45.1	**2.65**
Soft-NMS	40.8	**23.0**	43.4	53.2	**1.77**
SNIP	45.7	**29.3**	48.8	57.1	**1.56**
anchor-based one-stage					
YOLOv2	21.6	**5.0**	22.4	35.5	**4.32**
DSSD513	33.2	**13.0**	35.4	51.1	**2.55**
RetinaNet	39.1	**21.8**	42.7	50.2	**1.79**
anchor-free keypoint-based					
ExtremeNet	40.2	**20.4**	43.2	53.1	**1.97**
CenterNet	44.9	**25.6**	47.4	57.4	**1.75**
RepPoints	45.0	**26.6**	48.6	57.5	**1.69**
anchor-free center-based					
GA-RPN	39.8	**21.8**	42.6	50.7	**1.83**
FSAF	42.9	**26.6**	46.2	52.7	**1.61**
FCOS	43.2	**26.5**	46.2	53.3	**1.63**

outcomes in recent times. Nevertheless, scholarly attention has predominantly gravitated towards refining neural network designs and adjusting convolutional structures, inadvertently leading to suboptimal performance for small objects in comparison to their medium to large counterparts. This inherent imbalance in focus has, regrettably, impeded the overarching progress within the realm of object detection.

Generally speaking, the performance of small objects is limited, impacting the overall performance growth. As shown in Table 1, AP_s significantly lags behind AP, AP_m, and AP_l. Enhancing the performance of small objects, as discussed in [2] and [3], can lead to significant progress in general object detection.

The early explicit attempts to address the incongruity in detecting objects of different sizes include SSD [4] and FPN [5]. The use of a single-granularity vanilla convolution kernel (typically 3×3 or 5×5 [6, 7]) in the *backbone* restricts the feature extraction capability for objects of varying sizes. However, the analysis of datasets and consideration for the characteristics of different-size objects are often overlooked. There should be a greater focus on the analysis of datasets and the choice of kernel sizes for different-size objects in the *backbone*. To address this issue, we propose the **M**ulti-**g**ranularity **D**etector (MgD), featuring **M**ulti-granularity **F**eature **E**xtraction (MFE, or *stomach*) and **S**equential three-**w**ay **S**election (S3WS). The MgD is built on a reconstruction of network architectures and a redesign of evaluation functions at the surgical level.

To ameliorate the incongruity in detecting objects of various sizes, the Multi-granularity Feature Extraction (MFE) module incorporates customizable deformable convolution kernels tailored to different object sizes. This customization initiates with the adaptation of the kernel for small objects based on the backbone. Subsequently, scale factors k_1 and k_2 influence the size of deformable convolution kernels for medium and large objects. The MFE module applies three customizable deformable convolutions to three feature maps extracted from the backbone. Each feature map, with its associated customizable deformable convolutions, constitutes a stomach net, and three such stomach nets collectively form a "stomach" module. This modular approach contributes to a more adaptive and effective feature extraction process for objects of diverse sizes.

Moreover, addressing the challenge of an imbalance between positive and negative samples is crucial for enhancing detector performance. Achieving a balanced ratio, such as a 1:3 positive-to-negative sample ratio, significantly contributes to

detector efficacy[8]. In response to this challenge, we introduce the Sequential Three-way Selection (S3WS) module. Region proposals generated by the neural network undergo scoring by multiple evaluation functions, and these scores are input into the S3WS module. Region proposals x with an evaluation value $IoU_i(x)$ greater than α_i are classified as positive samples, those less than β_i as negative samples, and those falling between α_i and β_i as the boundary region[9, 10]. Additionally, region proposals within the boundary region undergo further classification until the stopping criterion is met. Positive samples are selected from both positive and boundary regions in a layered manner, based on multiple evaluation functions and two dynamic thresholds. In contrast, the selection of negative samples occurs in only one layer. The dynamic determination of the two thresholds, α_i and β_i, is facilitated by the evaluation function and the region proposals within the same batch size. This adaptive approach ensures the effective and context-aware selection of positive and negative samples throughout the network.

The primary contributions of this work can be summarized as follows:

- We assert that the detection results of small objects and the imbalance between positive and negative samples significantly constrain the overall detector performance.
- The MFE module, comprising multi-granularity deformable convolution kernels, is proposed to enhance the incongruity in detecting objects of different sizes. Simultaneously, the S3WS module is introduced to ameliorate the imbalance between positive and negative samples.
- To this end, the proposed MFE and S3WS modules can be seamlessly integrated into ConvNet detectors [11] and transformer-based detectors [12], yielding substantial improvements. Notably, SwinV2-G with MFE and SW3S (AP 63.1→64.0, AP/AP_s 1.97→1.42) surpasses other state-of-the-art results, albeit with a slightly larger model size.
- Our method, MgD (refer to Table11), outperforms all other state-of-the-art models on MS COCO[13] and enhances the contribution of small objects.

2. Related Work

Our work builds upon previous efforts in several domains: the analysis of datasets, reevaluation of backbone architectures and convolution kernels, and the redesigning of evaluation functions.

2.1. CNN and variant

The R-CNN series and YOLO series stand as archetypal embodiments of the two-stage model [14] and one-stage model [15], respectively, within the domain of object detection. The inaugural milestone in leveraging deep learning for object detection was marked by R-CNN [16]. Subsequent advancements, such as Fast R-CNN [16] and Faster R-CNN [17] (illustrated in Figure 1), laid the foundational framework for applying deep learning to object detection. YOLO introduces a more direct approach by regressing the bounding box's location and determining the bounding box's associated class, thereby reframing the object detection problem as a regression problem. Following this paradigm, numerous YOLO models [7, 18, 19] have been proposed, enhancing not only accuracy but also the computational speed of deep learning networks. However, these models overlook the nuanced distinctions between large and small objects in the dataset.

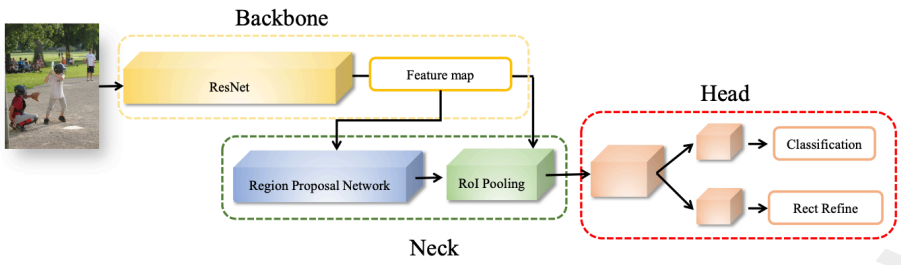

Figure 1: The main components of traditional object detection model. Most existing detection models mainly consist of *backbone*, *neck* and *head*. Note that we only show Faster R-CNN as an example.

2.2. *Backbone architectures*

The SSD [4] and FPN [5] are the first explicit attempts to solve the incongruity of different-size object detection results. These solutions did improve object detection results, however, they still ignored the data characteristics and statistical information of large and small objects. Afterwards, scholars prefered to improve the results of object detection by deepening or widening the neural network backbone (eg. AlexNet, GoogLeNet, and ResNet) without detailed analysis of the differences between large and small objects [20, 21].

2.3. *Convolution kernels*

Simultaneously, advancements in convolutional kernels [22] have unfolded. Notably, deformable convolution [23] introduces an offset variable to each sampled point's position within the convolution kernel. This innovation facilitates random sampling around the current position, liberating the convolution process from the constraints of the conventional regular grid points [24]. Additionally, dilated convolution strategically emphasizes the semantic information of local pixel blocks. By allowing each pixel to aggregate information from the surrounding blocks, it profoundly influences the granularity of segmentation outcomes [25].

2.4. *Small object detection*

Small object detection is a prominent focus in research, with efforts to improve performance. Traditional methods involve using high-resolution images or feature maps, but they come with substantial computational costs. Recent advancements aim to balance performance and efficiency. QueryDet, proposed by Yang et al. in [26], introduces a novel query mechanism to speed up object detectors based on feature pyramids. The process involves predicting approximate small object locations on low-resolution features and refining results using high-resolution features guided by coarse positions.

Gong Cheng's work in [27] conducts a thorough review of small object detection, catalyzing the development of Small Object Detection (SOD). Two datasets, SODA-D and SODA-A, focusing on driving and aerial scenarios, respectively, are introduced to further progress in this area.

Li et al.'s contribution in [28] presents a parallel multi-branch architecture with shared transformation parameters but different receptive fields. They use a scale-aware training scheme, sampling object instances with appropriate scales for training, to refine each branch's specialization. These influential works provide valuable inspiration for overcoming challenges in small object detection.

2.5. Imbalance between positive and negative samples

The escalating issue of imbalanced positive and negative samples, exacerbated by deeper neural networks and intricate convolutional structures [29], prompts the exploration of solutions from both data and algorithmic standpoints in machine learning. Techniques such as data augmentation, Online Hard Example Mining (OHEM), and Gradient Harmonizing Mechanism (GHM) have been employed. OHEM, as outlined in [30], identifies challenging examples based on input sample loss, emphasizing their impact on classification and detection during training with stochastic gradient descent.

Focal Loss [31] addresses sample imbalance by adapting classic cross-entropy loss. However, it relies on two intricate hyperparameters that demand substantial tuning and remains static, lacking adaptability to changing data distributions during training. To mitigate this, Li and Liu introduce GHM [32], a novel mechanism that harmonizes gradients to alleviate disharmonies. GHM's philosophy seamlessly integrates into both classification loss functions like cross-entropy (CE) and regression loss functions like smooth-L1 (SL1), offering a unique approach compared to Focal Loss, which focuses on confidence to attenuate losses. GHM, instead, mitigates losses based on sample size with a specified confidence level.

In [33], Oksuz and Baris conduct a comprehensive review of imbalance issues in object detection. To present a holistic perspective, they introduce a taxonomy outlining the problems and corresponding solutions to address them. Their approach is marked by a commitment to providing a thorough and detailed understanding of the problem landscape. To accomplish this, the authors introduce a comprehensive taxonomy that systematically categorizes the various challenges stemming from imbalance issues in object detection.

3. Methodology

The innovation of MgD are MFE and S3WS models: (1) the cores of MFE are multi-granularity deformable convolution layers to remedy poor result of small objects; (2) S3WS ameliorates the imbalance of positive and negative samples by selecting positive and negative samples in unequivalent way.

3.1. Analysis of the original dataset

Table 1 shows that the general performance of object detection is twice or three times more than that of small objects. In other words, the detection result of small objects limits the general performance. This is because in object detection, equal attention was paid to large, medium, and small objects, respectively, which means that researchers overlooked the analysis of data characteristics for different-size objects in the same dataset.

For the MS COCO 2017, Table 2 exhibits that the $\Gamma_{\#}$ of small, medium and large objects is almost the same. However, there are huge gaps between small, medium and large objects in $\Theta_{\#}$, $\Lambda_{\#}$, and $\Phi_{\#}$, which also leads to more focus on large objects. Previous methods prefer to randomly copy and paste small objects in the images to increase the occurrence of small objects. However, the improvement by this strategy is quite limited.

3.2. Multi-granularity deformable convolution layers

In this section, we design a **M**ulti-granularity **F**eature **E**xtraction module (MFE, or called *stomach*, extracting richer details, just like the stomach of an animal extracting rich nutrients from food.) by analyzing the origin dataset in detail. The multi-granularity deformable convolution layers consist of three feature maps released from *backbone* and the three customizable deformable convolution kernels. Each customizable deformable convolution kernel has its own modulation mechanism which

Table 2: Statistical information on labeled objects on MS COCO. $\Gamma_{\#}$ is the ratio of the number of # objects to the number of all objects, namely, $\Gamma_{\#} = \frac{\text{the number of \# objects}}{\text{the number of all objects}}$, where # = small, medium or large. $\Theta_{\#}$ is the ratio of the total area of # objects to the total area of all objects, namely, $\Theta_{\#} = \frac{\text{the total area of \# objects}}{\text{the total area of all objects}}$, where # = small, medium or large. $\Lambda_{\#}$ is the ratio of the number of images containing # objects to the total number of images, namely, $\Lambda_{\#} = \frac{\text{the number of images containing \# objects}}{\text{the total number of images}}$, where # = small, medium or large. $\Phi_{\#}$ is the average area of # objects (number of pixels), where # = small, medium or large.

Size	$\Gamma_{\#}$	$\Theta_{\#}$	$\Lambda_{\#}$	$\Phi_{\#}$
large	33.97%	93.44%	91.22%	8995.63
medium	34.90%	5.99%	64.72%	3201.15
small	31.13%	0.57%	43.54%	714.23

is realized by a weighted convolution. Meanwhile, the RoI pooling layer changes accordingly due to modulation mechanism. The deformable convolution is expressed as follows:

$$y(p) = \sum_{k=1}^{K} w_k \cdot x\left(p + p_k + \Delta p_k\right) \cdot \Delta m_k, \tag{1}$$

where Δp_k and Δm_k are the learnable offset and modulation scalar for the k th location, respectively. $y(p)$ represents the output feature y in the position p. The modulation scalar Δm_k lies in the range [0, 1]. For detailed settings, please refer to [34].

Considering the diverse aspect ratios of objects, we use the average area to define the ratio of kernel sizes (k_1 and k_2). The following formulas are used to determine the value of k_1 and k_2 :

$$\frac{KS_{\text{small}}}{KS_{\text{medium}}} = \sqrt{\frac{Aa_{\text{small}}}{Aa_{\text{medium}}}} = \frac{1}{k_1} \tag{2}$$

$$\frac{KS_{\text{medium}}}{KS_{\text{large}}} = \sqrt{\frac{Aa_{\text{medium}}}{Aa_{\text{large}}}} = \frac{1}{k_2} \tag{3}$$

where KS_{small} means the kernel size of small objects in single dimension, and Aa_{small} is the average area of small objects. One has the same explanation for KS_{medium}, Aa_{medium}, KS_{large} and Aa_{large}. With the information show in Table 2, we calculate that $k_1 \approx 2.11$, $k_2 \approx 1.45$.

Figure 2 exhibits that in one stomach net, three customizable deformable convolution kernels are utilized to convolute each feature map obtained from the last three convolution layers in *backbone*. The size of the deformable convolution kernel is the key to extract the feature of small objects, which also depends on the specifics of the previous *backbone*. Generally, the size of the deformable convolution kernel for small objects cannot be larger than that of the convolution kernel in the last layer of *backbone*. In this section, we perform the following settings: $KS_{\text{small}} = 3$, $KS_{\text{medium}} = 5$, $KS_{\text{large}} = 7$.

Three stomach nets form a new module *stomach*. This new module works like the human stomach, extracting the feature map from the upstream, and provides *neck* with more accurate and detailed data according to the size of different objects in Figure 3.

Deformation convolution is relative to the concept of standard convolution. In the standard convolution operation, the area of action of the convolution kernel is always a rectangular area with the size of the standard convolution kernel around the center point.

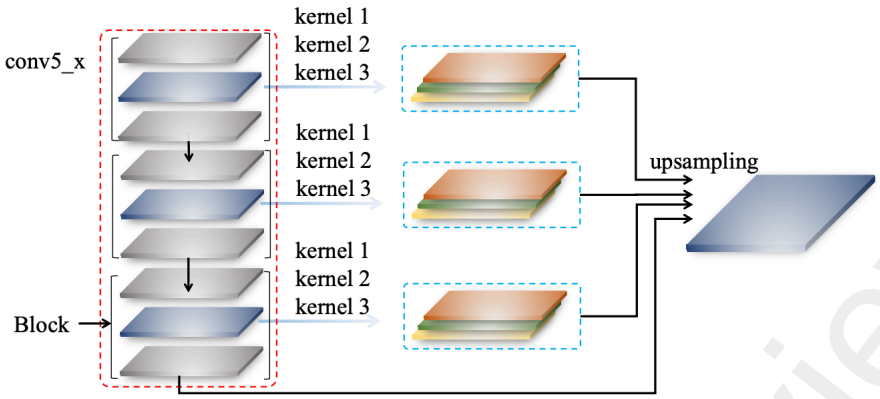

Figure 2: *Stomach net*. Here, we adopt ResNet101 as an example. In con5_x layer of ResNet101, every feature map after kernel 3×3 (blue ones in three blocks) are utilized with multi-granularity kernels (kernel 1, kernel 2, and kernel 3).

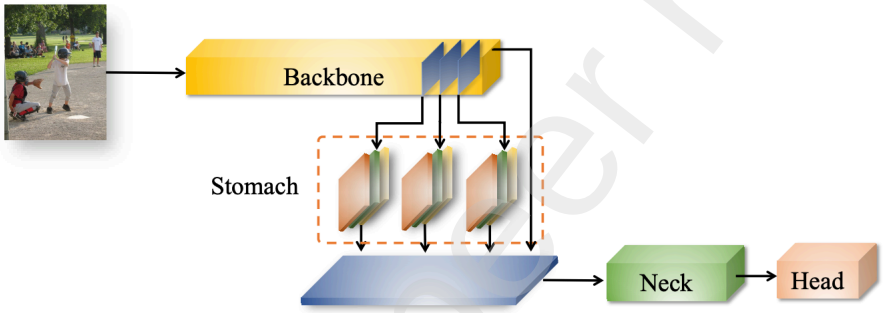

Figure 3: *Stomach*.

These deformable convolutions are fixed in deformable detr. Our deformable convolutions are based on the statistical characteristics of large, small and medium objects in the dataset, which are customized and varies according to dataset.

Deformable detr innovations focus on combinations of standard deformable convolution and transformer. One of our innovations is that we propose that the convolution in the backbone should not be static, but should be customized according to the statistical information of dataset.

3.3. Sequential three-way selection for region proposals

The prevailing detectors grapple with a pronounced imbalance between positive and negative samples. In this section, we introduce the Sequential Three-Way Selection (S3WS) module, amalgamating the concept of sequential three-way decision with a selection module, to rectify the imbalance inherent in positive and negative samples. The sequential three-way decision involves a sequence of three-way decisions. The fundamental concept of a three-way decision revolves around partitioning a set of objects into positive, negative, and boundary regions, guided by evaluation functions and decision parameters α and β. Objects in the positive and negative regions are subjected to definitive decisionsacceptance and rejection, respectively. Objects residing in the boundary region undergo an additional iteration of the three-way decision process [9, 10].

Let U be a set of region proposals and I a set of different evaluation functions, i.e., $I = \{IoU_1, IoU_2, IoU_3, ...\}$, where

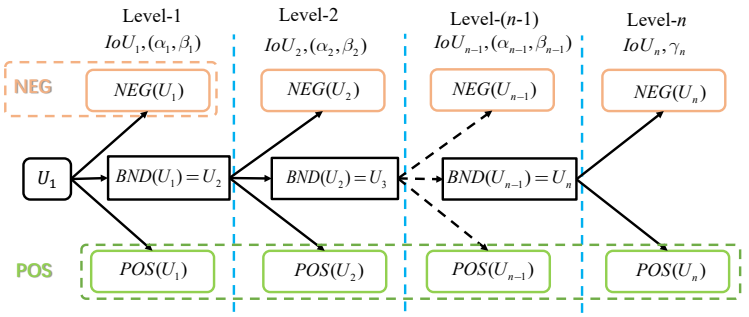

Figure 4: Sequential three-way selection module. POS means positive sample set and NEG means negative sample set. $POS = POS(U_1) \cup \cdots \cup POS(U_n)$ and $NEG = NEG(U_1)$.

$IoU_i = tIoU$[2], $GIoU, CIoU, DIoU, or\ CDIoU$. A S3WS module is shown in Figure 4. At *Level-i*, we choose a certain IoU function as the evaluation function. The decision parameters α_i and β_i are dynamically determined by the following formulas:

$$\begin{aligned} \alpha_i &= \frac{1}{m}\sum_{j=1}^{m} IoU_i(x) \\ \beta_i &= \alpha_i - \sqrt{\frac{\sum_{j=1}^{m}\left(IoU_i(x)-\alpha_i\right)^2}{m}} \\ i &= 1, 2, ..., n-1 \end{aligned} \tag{4}$$

where i means ith level, while α and β mean positive threshold and negative threshold. And x means region proposals in a set (eg. U). m means the number of region proposals in a set.

At the initial level, namely, *Level-1*, the starting universe U_1 is just the whole universal set U. U_1 is divided into three regions on the basis of the decision function IoU_1 and the pair of thresholds (α_1, β_1):

$$\begin{aligned} \text{POS}\,(U_1) &= \{x \in U_1 \mid IoU_1(x) \geq \alpha_1\} \\ \text{BND}\,(U_1) &= \{x \in U_1 \mid \beta_1 < IoU_1(x) < \alpha_1\} \\ \text{NEG}\,(U_1) &= \{x \in U_1 \mid IoU_1(x) \leq \beta_1\} \end{aligned} \tag{5}$$

The boundary region $BND(U_1)$ is then regarded as the universe U_2 based on which the next stage of three-way selection proceeds. The universe U_2 is then divided into the following three regions:

$$\begin{aligned} \text{POS}\,(U_2) &= \{x \in U_2 \mid IoU_2(x) \geq \alpha_2\} \\ \text{BND}\,(U_2) &= \{x \in U_2 \mid \beta_2 < IoU_2(x) < \alpha_2\} \\ \text{NEG}\,(U_2) &= \{x \in U_2 \mid IoU_2(x) \leq \beta_2\} \end{aligned} \tag{6}$$

where IoU_2 is a new evaluation function and (α_2, β_2) is the pair of decision parameters of Level-2.

The boundary region $BND(U_2)$ is then regarded as the universe U_3. The same procedure goes on for universes U_3, U_4, $\cdots$ until U_{n-1}. For the universe U_n which is $BND(U_{n-1})$, a two-way decision strategy is adopted based on IoU_n and the threshold γ_n :

[2]tIoU means traditional intersection over union function, namely, tIoU = $\frac{A \cap B}{A \cup B}$. In the experiments, tIoU is expressed as IoU.

$$\begin{aligned} \mathrm{POS}\left(U_{n-1}\right) &= \{x \in U_{n-1} \mid IoU_n(x) \geq \gamma_n\} \\ \mathrm{NEG}\left(U_{n-1}\right) &= \{x \in U_{n-1} \mid IoU_n(x) < \gamma_n\} \end{aligned} \tag{7}$$

where γ_n= 0.5, 0.75, or 0.95 and 0.5 is the most common option. Naturally, the classification loss uses the original function. Due to our serialized employ of multiple IoUs for regressing, the regression loss function will be expressed as:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{IoU}_1} + \mathcal{L}_{\text{IoU}_2} + \cdots + \mathcal{L}_{\text{IoU}_n}. \tag{8}$$

4. Experiments

In this section, we commence by delineating the datasets and hardware specifications underpinning our experimentation. Subsequently, we expound upon the nuanced implementation details of the experiment, incorporating comprehensive ablation studies on MFE and S3WS modules. Finally, we embark on a discerning comparative analysis, positioning our method against contemporary state-of-the-art approaches.

Settings. The experimental paradigm unfolds within the realms of MS COCO 2017 and PASCAL VOC 2012 datasets, leveraging the computational prowess of 2 GeForce RTX 3090 and 2 Tesla V100 PCIe 32GB. All model implementations within the PyTorch or TensorFlow frameworks adhere to canonical configurations, devoid of any idiosyncratic embellishments. Four preeminent object detection frameworksATSS[35], Faster RCNN[17], Swin Transformer[36, 37], and DETRs[38]serve as benchmarks for ablation studies and comparative evaluations.

Dataset. The experimental purview extends over the COCO 2017 detection datasets, encompassing 118k training images, 5k validation images, and 20K test-dev images. The validation set facilitates meticulous ablation studies, while the test-dev set provides the canvas for a comprehensive system-level comparison. Each image is meticulously annotated with bounding boxes and panoptic segmentation, featuring an average of 7 instances during training, exhibiting a spectrum from diminutive to expansive dimensions.

PASCAL VOC 2012 furnishes a repository of 17,125 images of varied dimensions, spanning four categories: people, animals, vehicles, and indoor furniture, along with subcategories, culminating in a total of 20 distinct image categories. The training dataset comprises annotated images, each equipped with bounding box annotations and object class labels corresponding to one of the twenty classes. Instances of multiple objects from diverse classes coexisting within a single image underscore the dataset's complexity. The dataset is judiciously partitioned, with a 50% allocation for training/validation and an equivalent 50% for testing, ensuring parity in the distribution of images and objects across sets.

Training. The training regimen for the MgD model integrates both Adamw and SGD optimizers, transitioning from Adamw to SGD in the ultimate stages of training. The backbone features EfficientNetD3/D5/D7, and the learning rate for the backbone is meticulously set at 2^{-5}. The training adheres to the established DETR standards, incorporating the ImageNet-pretrained model from TORCHVISION for the backbone, with batchnorm layers held constant. Transformer parameters undergo initialization via the Xavier scheme, accompanied by a weight decay set at 10^{-4}. The configurations of ATSS and Faster RCNN align with the specifications detailed in [35]. During training, the infusion of horizontal flipping and scale jittering [0.1, 2.0] introduces stochasticity, randomly resizing images within the range of 0.1x to 2.0x of the original size before cropping. The evaluation phase is marked by the application of soft-NMS.

The evaluation of MgD on COCO 2017 detection datasets, featuring 118K training images, unfolds under the aegis of an SGD optimizer characterized by momentum 0.9 and weight decay 4^{-5}. The learning rate trajectory is characterized by a linear ascent from 0 to 0.16 in the inaugural training epoch, subsequently subject to an annealing process via the cosine decay rule. Synchronized batch norm is strategically introduced post each convolutional operation, featuring a batch norm decay of 0.99 and an epsilon of 1^{-3}.

4.1. Ablation studies on MFE

Naturally, the released feature maps in different positions of *backbone* have different effects on final performance. Figure 5 displays the different positions of *stomachs*. From Tables 3 and 4 , one can conclude that *postorder stomach* module improves detection results most effectively. Moreover, along with the movement of *stomach* to the front of *backbone*, the detection results decrease rapidly and reach the lowest for *homorder stomach*.

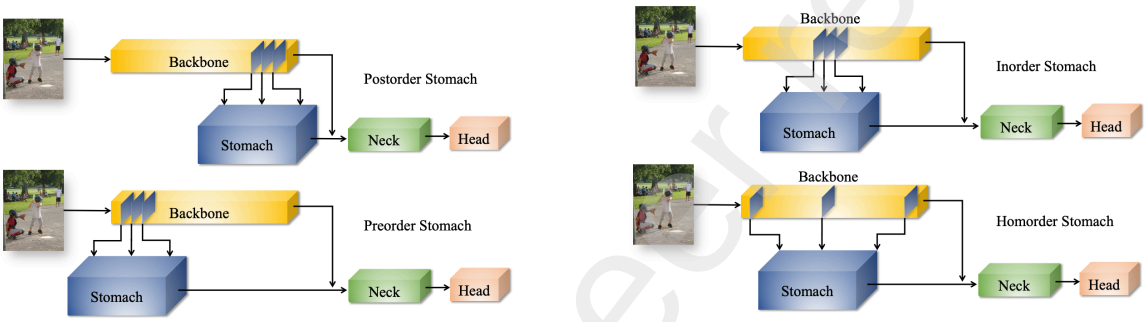

Figure 5: *Stomaches* in different positions. *Postorder stomach*: *stomach* in the last three layers of *backbone*; *inorder stomach*: in the middle three layers; *preorder stomach*: in the first three layers; *homorder stomach*: *stomach nets* are homogeneously distributed in *backbone*.

In previous experiments, we regarded the convolution kernel adapted to small objects as the **B**asic **C**onvolution **K**ernel (BCK), and determined the convolution kernel size of medium-sized and large objects based on BCK. After a number of comparative experiments, we repeatedly adjusted BCK , and found the following reasons to explain this decline:

- According to Formula 2 and 3, we determine the convolution kernel size of medium and large objects based on BCK, which slightly restricts the extraction of large object features in *backbone*. This influence will be amplified with the continuous forward movement of *stomach* module until it moves to the front end of *backbone*.
- After images are processed by *stomach*, the feature maps can meet the input requirements of *neck* only through *matching operations* such as up sampling, down sampling, or deconvolution. This *matching operation* will gradually split the semantic information of objects according to the aggravation of size difference between the upstream and downstream feature maps.

It is clear that MFE shows significant differences for large, medium, and small objects (see Tables 3 and 4). Compared to AP, AP_m, and AP_l, AP_s gets the maximum gain with *Postorder stomach* on MS COCO *validation* or *test-dev* set. At ATSS, AP_s gain (+5.4) is 3.8 times that of AP gain (FCOS : 3.1 times, Faster RCNN: 4.4 times, and MgD: 2.1 times). The changes in AP_m and AP_l are moderate and understandable (ATSS: AP_m +2.4, AP_l +0.1; FCOS: AP_m +2.7, AP_l +0.0 ; Faster RCNN: AP_m +2.7, AP_l -0.1; MgD: AP_m +1.1, AP_l +0.7), compared to AP_s. On PASCAL VOC *validation* or *test* set, this improvement of MFE is more pronounced.

Table 3: Detection results (%) with AP_s, AP_m, AP_l on MS COCO *validation* or *test-dev* set. All modules are on trainval35k. ATSS backbone: ResNet-101; FCOS backbone: ResNeXt-64x4d-101; MgD backbone: EfficientNet-2. Origin part is on *test-dev* set and the other parts are on *validation* set. The numbers with + in parentheses indicate the improvement of the results.

Method		AP	AP_s	AP_m	AP_l
ATSS	origin	43.6	**26.1**	47.0	53.6
	post	45.0(+1.40)	**31.5**(+5.4)	49.4	53.7
	in	40.5	**25.5**	46.1	50.6
	pre	35.7	**18.9**	40.5	45.9
	home	29.9	**12.4**	30.4	42.1
FCOS	origin	43.2	**26.5**	46.2	53.3
	post	45.0(+1.8)	**32.0**(+5.5)	48.9	53.3
	in	41.0	**25.9**	44.7	50.1
	pre	38.4	**20.4**	38.9	48.1
	home	30.3	**17.6**	31.6	41.1
Faster RCNN	origin	36.0	**18.2**	39.0	48.2
	post	37.7(+1.7)	**25.7**(+7.5)	41.7	48.1
	in	33.3	**23.4**	40.5	45.6
	pre	27.0	**12.5**	28.4	40.0
	home	21.0	**8.9**	22.9	36.7
MgD	origin	40.4	**25.7**	43.7	50.1
	post	42.1(+1.9)	**29.7**(+4.0)	44.6	50.8
	in	40.0	**28.4**	43.4	47.5
	pre	32.1	**15.0**	33.0	40.1
	home	28.0	**12.0**	29.1	35.4

MFE (with backbone) proposes more feature information of small objects, compared to the original backbone, which provides more detailed information for subsequent detection heads. The MFE module significantly improves the performance of AP_s without harming AP, AP_m, and AP_l.

4.2. Ablation studies on diffusion

For a data acquisition point (or a grid) of a deformable convolution, we call the shifting of the grid sampling locations an *offset*. For the overall deformable convolution, the offset of all data acquisition points causes the acquisition area of the overall convolution to spread outward. We call this *diffusion*.

We designed a series of comparative experiments with different diffusion levels shown in Figure 6. Several classical models run on *postorder stomach* with *diffusion level-1*, *diffusion level-2*, *diffusion level-3*, and *free diffusion*, respectively. *Free diffusion* means that instead of specifying a hard offset distance for the convolution kernel, the deep learning network automatically learns the offset distance.

The detection results were recorded in Figure 7. As the diffusion level increases, the model performance does not increase but decreases rapidly. The best results were obtained with free diffusion *stomach* modules, about 1∼2% higher than the original performance.

In the experiment, we tried several density options. If we set 1 or 2 layers in *stomach*, the performance cannot be improved significantly. When we set 3 layers in *stomach*,, the detection performance has been significantly improved. However, we have set more than 3 layers then the results are similar to the detection performance of 3 layers.

4.3. Ablation studies on S3WS module

Regarding to choose the evaluation functions in S3WS, we analyzed and classified the major evaluation functions at first. We classify evaluation functions into three main categories:

Table 4: Detection results (%) with AP_s, AP_m, AP_l on PASCAL VOC *validation* or *test* set. FCOS backbone: ResNeXt-64x4d-101; MgD backbone: EfficientNet-2. Origin part is on *test-dev* set and the other parts are on *validation* set. The numbers with + in parentheses indicate the improvement of the results.

Method		AP	AP_s	AP_m	AP_l
FCOS	origin	75.2	**36.5**	79.2	82.4
	post	77.6(+2.4)	**42.0**(+5.5)	78.2	85.6
	in	71.3	**36.8**	74.7	80.1
	pre	65.3	**32.5**	68.2	70.1
	home	56.3	**25.9**	61.5	61.9
Faster RCNN	origin	73.8	**38.2**	75.0	79.2
	post	77.7(+3.9)	**45.7**(+7.5)	79.7	81.9
	in	69.1	**33.4**	70.5	77.6
	pre	62.6	**32.5**	68.4	72.0
	home	58.7	**29.5**	59.2	63.7
MgD	origin	75.4	**35.4**	76.9	80.1
	post	78.4(+3.0)	**38.5**(+3.1)	75.6	81.7
	in	70.0	**34.9**	72.0	77.5
	pre	62.1	**30.0**	63.4	65.9
	home	58.0	**22.9**	59.0	55.4

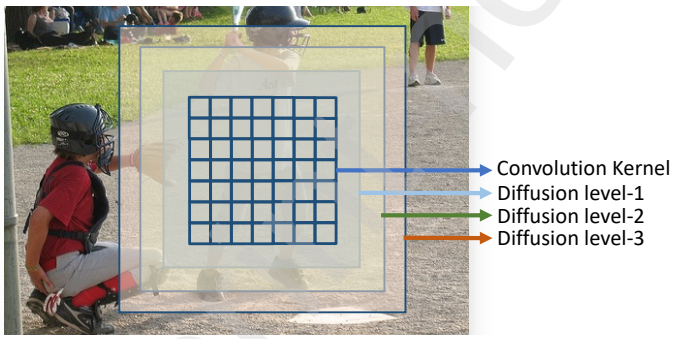

Figure 6: Deformable convolution kernel with different diffusion. The blue grids represent the deformable convolution kernel. The blue translucent rectangular box symbolizes the convolution kernel diffusing outward by one pixel unit; the green one symbolizes two pixel units; the red one symbolizes three pixel units.

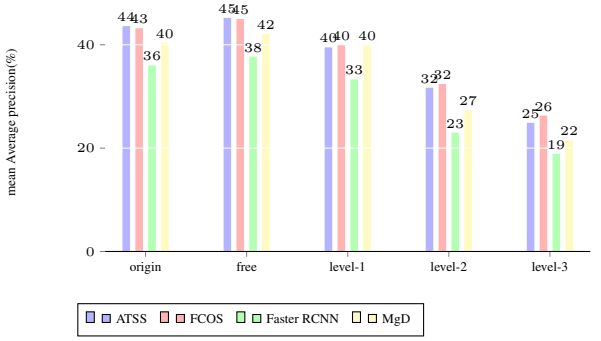

Figure 7: Detection results (%) with different *diffusion levels* on MS COCO *validation* set. Level-1 = *diffusion level-1*; Level-2 = *diffusion level-2*; Level-3 = *diffusion level-3*; Free = *free diffusion*.

Table 5: Detection results (%) with different S3WS modules on MS COCO *validation* set. All modules are on trainval35k. ATSS backbone: ResNet-101; MgD backbone: EfficientNet-2. IoU-GIoU-CDIoU means that IoU is selected as the evaluation function for Level-1, GIoU for Level-2 and CDIoU for Level-3. Bold fonts indicate the best performance. AP/AP_s is proportion.

S3WS combinations	ATSS		Faster RCNN		MgD	
	AP	AP/AP_s	AP	AP/AP_s	AP	AP/AP_s
original	43.6	1.67	36.0	1.98	40.4	1.57
IoU-GIoU	43.8	1.56	36.9	1.80	41.0	1.51
IoU-CDIoU	43.9	1.55	36.8	1.88	40.9	1.53
IoU-CIoU	43.8	1.55	36.7	1.85	40.8	1.53
IoU-GIoU-CDIoU	**44.0**	1.45	**37.1**	1.65	**41.1**	1.40
IoU-GIoU-CIoU	43.9	1.44	**37.1**	1.64	40.9	1.40
IoU-GIoU-DIoU	43.9	1.45	37.0	1.63	40.6	1.39
IoU-GIoU-CIoU-DIoU	43.5	1.42	36.2	1.60	40.4	1.34

Table 6: Detection results (%) with different S3WS modules on PASCAL VOC *validation* set. MgD backbone: EfficientNet-2. IoU-GIoU-CDIoU means that IoU is selected as the evaluation function for Level-1, GIoU for Level-2 and CDIoU for Level-3. Bold fonts indicate the best performance. AP/AP_s is proportion.

S3WS combinations	Faster RCNN		MgD	
	AP	AP/AP_s	AP	AP/AP_s
original	73.8	2.18	80.4	1.76
IoU-GIoU	75.9	1.80	81.0	1.51
IoU-CDIoU	76.8	1.88	80.9	1.51
IoU-CIoU	76.7	1.85	81.1	1.52
IoU-GIoU-CDIoU	**77.3**	1.65	**81.1**	1.40
IoU-GIoU-CIoU	77.1	1.64	80.7	1.40
IoU-GIoU-DIoU	77.0	1.63	81.0	1.39
IoU-GIoU-CIoU-DIoU	72.0	1.60	79.3	1.33

- Type 1: focusing on the measure overlapping area (eg. IoU);
- Type 2: focusing on the ratio of overlapping area to unoverlapping area (eg. GIoU);
- Type 3: focusing on a measure of difference, sometimes understood as centroid distance and aspect ratio (eg. CDIoU, CIoU, DIoU).

Then, in each level of S3WS module, a certain type of evaluation function is applied. Along with the increase of the level of S3WS, the performance of detectors are continuously improved (see in Table 5 and 6) until 3 levels. The experiments testified that the combination *IoU-GIoU-CDIoU* achieves the best result. The combinations obtain representative results are exhibited in Table 5 and 6 for different combinations of evaluation functions. The three-level S3WS modules significantly improve the results of detectors. When S3WS exceeds four levels, it not only brings no improvement in results, but also leads to extremely slow convergence of loss functions, which will cause runtime more than four-month. The rate of positive and negative samples decreasing gradually, when the numbers of level go up: ATSS 1:11 (original)→1:9 (2 levels)→1:7 (3 levels)→1:6(4 levels); Faster RCNN 1:200→1:150→1:120→1:100; MgD 1:20→1:12→1:7→1:6.

Based on the above experiments, we conclude that the same type of evaluation functions form a pairwise antagonistic relationship within the detection model. The detection models with S3WS modules more than three levels cannot reach the minimum value of multiple evaluation functions. As a result, the feedback mechanism feeds large values to the backprogration, which eventually leads to the model failing to converge.

4.4. Object Detection on VisDrone

Experimental Configuration. Ensuring a rigorous basis for comparison, we meticulously adhere to a standardized experimental setup across diverse methodologies. This entails the adoption of consistent configurations, encompassing multi-scale training, the utilization of the AdamW optimizer initialized with a learning rate of 0.00001, and the inclusion of a weight decay set at 0.05 within the mmdetection framework. Furthermore, our models are initialized with the ImageNet-22K pre-trained model, providing a comprehensive foundation for system-level evaluations.

Dataset Details. The VisDrone dataset constitutes a rich repository comprising 400 video clips, totaling 265,228 frames and 10,209 static images. Captured through an array of drone-mounted cameras, the dataset encapsulates diverse spatial locales, environmental conditions, and objects classified into 10 distinct categories. Each frame undergoes meticulous manual annotation, resulting in over 2.6 million precisely delineated bounding boxes or points of interest, characterizing entities ranging from pedestrians and cars to bicycles and tricycles.

Visdrone, distinguished as a professional-grade dataset expressly tailored for drone-centric applications, is particularly noteworthy for its abundance of samples featuring small objects. Remarkably, in tackling the challenges posed by this intricate dataset, MgD exhibits commendable performance without an unwarranted escalation in computational cost and time.

Table 7 compares our best results with those of previous state-of-the-art models in VisDrone-DET 2020 and 2021 Challenge. MgD achieves $41.6AP, 65.5AP_{50}$, and $43.4AP_{75}$ on VisDrone, surpassing all previous best results in Table 7.

4.5. Comparison

Via the aforementioned ablation studies, we have ascertained the optimal configuration for MFE and S3WS. To corroborate the efficacy of these modules, we conducted additional comparative experiments on prominent models. The outcomes are delineated in Table 8. The incremental incorporation of MFE and S3WS consistently yields notable enhancements in the detectors' performance.

From Table 10, we can see that our model MgD has significant advantages in detection performance, FPS, model size, and testing time. MgD achieves similar detection performance with around $1/10$ model size, $7/10$ testing time, and $1.5\times$ FPS. Because various methods also use various non customized convolutional kernels and IoU loss functions. So MFE and SW3S will not incur any additional computational costs or model size.

Evaluation against Conventional ConvNets. The performance metrics of ATSS and Faster RCNN with/without the integration of MFE and SW3S are presented in the left segment of Table 9. Results incorporating MFE and SW3S exhibit a notable increase of +1.1/+2.0 AP and a decrease of -0.36/-0.55 AP/AP_s compared to their counterparts without these modules. Interestingly, the model size and inference speed remain largely unaffected.

Evaluation against Transformer-Based Approaches. The right portion of Table 9 showcases the performance of Swin Transformer, Swin Transformer V2, DETR, and UP-DETR with/without MFE and SW3S. It is evident that transformer-based methodologies encounter challenges related to suboptimal results for small objects and an imbalance between positive and negative samples. The incorporation of MFE and SW3S yields substantial improvements, with results featuring +1.2/+1.3 AP and -0.47/-0.49 AP/AP_s increases/decreases compared to configurations lacking these modules. Notably, SwinV2-G with MFE and SW3S (AP 63.1→64.0, AP/AP_s 1.97→1.42) surpasses prevailing state-of-the-art outcomes.

Evaluation against Prior State-of-the-Art Models. Introducing the novel MgD detector, equipped with MFE and S3WS modules after the EfficientNet, reveals superior performance on the MS COCO dataset in Table 11. MgD not only outperforms

Table 7: System-level comparison (%) of MgD on VisDrone-DET 2020[39] and 2021[40].

Method	AP	AP_{50}	AP_{75}
VisDrone-DET 2020[39]			
DroneEye2020 (A.4)	34.57	58.21	35.74
TAUN (A.5)	34.54	59.42	34.97
CDNet (A.6)	34.19	57.52	35.13
CascadeAdapt (A.7)	34.16	58.42	34.5
HR-Cascade++ (A.9)	32.47	55.06	33.34
MSC-CenterNet (A.11)	31.13	54.13	31.41
CenterNet+ (A.12)	30.94	52.82	31.13
ASNet (A.13)	29.57	52.25	29.37
CN-FaDhSa (A.14)	28.52	49.5	28.86
HRNet (A.15)	27.39	49.9	26.71
DMNet (A.16)	27.33	48.44	27.31
HRD-Net (A.17)	26.93	45.45	27.77
PG-YOLO (A.18)	26.05	49.63	24.15
EFPN (A.19)	25.27	48.18	23.37
CRENet (A.20)	25.16	44.38	24.57
Cascade R-CNN++ (A.21)	24.66	43.53	24.71
HR-ATSS (A.22)	24.23	41.84	24.43
CFPN (A.23)	22.85	42.33	21.88
Center-ClusterNet (A.24)	22.72	41.45	22.13
HRC (A.26)	21.23	43.56	18.39
IterDet (A.27)	20.42	36.73	20.25
GabA-Cascade (A.29)	18.85	33.60	18.66
VisDrone-DET 2021 [40]			
DBNet	39.4	65.4	41.0
SOLOer	39.4	63.9	40.8
Swin-T	39.4	63.9	40.8
TPH-YOLOv5	39.1	62.8	41.3
VistrongerDet	38.7	64.2	40.2
EfficientDet	38.5	63.2	39.5
DroneEye2020	34.5	58.2	35.7
Cascade R-CNN	16.0	31.9	15.0
DroneEye2020	34.57	58.21	35.74
DPNet-ensemble	37.3	62.0	39.1
MgD (EfficientNet-D3)	**41.6**	**65.5**	**43.4**

Table 8: Detection results (%) on MS COCO *test-dev* set or *validation* set. Bold fonts indicate the best performance. Swin Transformer backbone: Swin-L(HTC++), multi-scale testing. UP-DETR[38] backbone: ResNet50. The numbers with + in parentheses indicate the improvement of the results. The numbers with - in parenthesess indicate that the contribution of small object detection results is increasing. AP/AP_s is proportion.

Method	MFE	S3WS	$AP(\%)$	AP/AP_s
ATSS			43.6	1.67
	✓		45.0	1.42
	✓	✓	45.3(+1.7)	1.40(-0.27)
Faster RCNN			36.0	1.98
	✓		37.7	1.47
	✓	✓	38.0(+2.0)	1.42(-0.56)
Swin-Transformer			58.7	1.89
	✓		59.4	1.60
	✓	✓	59.6(+0.9)	1.54(-0.35)
UP-DETR			42.8	2.06
	✓		43.4	1.90
	✓	✓	43.8(+1.0)	1.81(-0.25)
MgD			40.4	1.57
	✓		42.1	1.41
	✓	✓	42.5(+2.1)	**1.40**(-0.17)

Table 9: Detection results (%) on MS COCO *validation* set. In *w.* item, ✓means that with MFE and SW3S modules. Swin-Tran means Swin Transformer and SwinV2-G (HTC++) with multi-scale testing. DETRs means DETR and UP-DETR with 300 epochs. Table 9 shows detector results from Detectron2 Model Zoo or MMDetection Model Zoo. AP/AP_s is proportion.

Method	Backbone	w.	AP	AP/AP_s	Method	Backbone	w.	AP	AP/AP_s
ATSS	ResNeXt-32x8d-101		45.1	1.66	Swin-Tran	Swin-S (Cascada Mask)		51.8	1.82
		✓	46.2	1.42			✓	52.4	1.60
	ResNet-101-DCN		46.3	1.72		Swin-B (HTC++)		56.4	1.97
		✓	47.4	1.43			✓	57.6	1.50
	ResNeXt-64x4d-101 -DCN		47.7	1.78		SwinV2-G (HTC++)		63.1	1.97
		✓	48.8	1.42			✓	**64.0**	**1.42**
Faster RCNN	VGG-16		36.0	1.98	DETRs	ResNet-50 (Supervision CNN)		40.8	2.27
		✓	38.0	1.42			✓	41.9	1.89
	ResNet-50		37.2	2.00		ResNet-50 (SwAV CNN)		42.1	2.37
		✓	38.4	1.50			✓	43.4	2.00
	ResNet-101		39.5	2.10		ResNet-50 (UP-DETR)		42.8	2.54
		✓	41.0	1.55			✓	43.7	2.11

Table 10: Detection results (%), FPS, model size, and testing time on MS COCO *validation* set.

Method	AP	FPS	model size	time/image
ATSS (ResNet-101)	43.6	9.0	196M	57ms
ATSS (ResNet-101) + MFE & SW3S	45.3	9.0	196M	57ms
FCOS (ResNeXt-64X4d-101)	43.2	–	345M	112ms
FCOS (ResNeXt-64X4d-101)+ MFE	45.0	–	345M	112ms
Faster RCNN (ResNet-50)	36.0	10.7	160M	–
Faster RCNN (ResNet-50) + MFE & SW3S	38.0	10.7	160M	–
MgD (EfficientNet-2)	40.4	**13.4**	**32.9M**	78ms
MgD (EfficientNet-2) + MFE & SW3S	**42.5**	**13.4**	**32.9M**	78ms

Table 11: Detection results (%) on MS COCO *test-dev* set or *validation* set. Bold fonts indicate the best performance. The red font indicates the best AP/AP_s, indicating that the contribution of small object detection has significantly improved the results of general object detection. AP/AP_s is proportion. FCOS + SaAA means FCOS + Scale-aware AutoAug.

Method	Data	Backbone	AP	AP_s	AP_m	AP_l	AP/AP_s
anchor-based two-stage							
MLKP	trainval35k	ResNet-101	28.6	10.8	33.4	45.1	2.65
R-FCN[14]	trainval	ResNet-101	29.9	10.8	32.8	45.0	2.76
CoupleNet	trainval	ResNet-101	34.4	13.4	38.1	50.8	2.57
TDM[41]	trainval	ResNet-v2-TDM	36.8	16.2	39.8	52.1	2.27
DeepRegionlets	trainval35k	ResNet-101	39.3	21.7	43.7	50.9	1.84
FitnessNMS	trainval	DeNet-101	39.5	18.9	43.5	54.1	2.09
DetNet[42]	trainval35k	DetNet-59	40.3	23.6	42.6	50.0	1.71
soft-NMS	trainval	ResNet-101	40.8	23.0	43.4	53.2	1.77
SOD-MTGAN[43]	trainval35k	RerNet-101	41.4	24.7	44.2	52.6	1.68
anchor-based one-stage							
YOLOv2[7]	trainval35k	DarkNet-19	21.6	5.0	22.4	35.5	4.32
SSD512[4]	trainval35k	VGG-16	28.8	10.9	31.8	43.5	2.64
STDN513[44]	trainval	DenseNet-169	31.8	14.4	36.1	43.4	2.21
DES512[45]	trainval35k	VGG-16	32.8	13.9	36.2	47.5	2.36
DSSD513[22]	trainval35k	ResNet-101	33.2	13.0	35.4	51.1	2.55
RFB512-E[46]	trainval35k	VGG-16	34.4	17.6	37.0	47.6	1.95
PFPNet-R512	trainval35k	VGG-16	35.2	18.7	38.6	45.9	1.88
RefineDet512	trainval35k	ResNet-101	36.4	16.6	39.9	51.4	2.19
RetinaNet	trainval35k	ResNet-101	39.1	21.8	42.7	50.2	1.79
anchor-free center-based							
GA-RPN[47]	trainval35k	ResNet-50	39.8	21.8	42.6	50.7	1.83
FoveaBox[48]	trainval35k	ResNeXt-101	42.1	24.9	46.8	55.6	1.69
FSAF[49]	trainval35k	ResNeXt-64x4d-101	42.9	26.6	46.2	52.7	1.61
FCOS[50]	trainval35k	ResNeXt-64x4d-101	43.2	26.5	46.2	53.3	1.63
anchor-free keypoint-based							
ExtremeNet[51]	trainval35k	Hourglass-104	40.2	20.4	43.2	53.1	1.97
CenterNet-HG[52]	trainval35k	Hourglass-104	42.1	24.1	45.5	52.8	1.75
Grid R-CNN	trainval35k	ResNeXt-101	43.2	25.1	46.5	55.2	1.72
CornerNet-Lite	trainval35k	Hourglass-54	43.2	24.4	44.6	57.3	1.77
CenterNet[53]	trainval35k	Hourglass-104	44.9	25.6	47.4	57.4	1.75
RepPoints[54]	trainval35k	ResNeXt-101-DCN	45.0	26.6	48.6	57.5	1.69
recent excellent models							
ATSS[35]	trainval35k	ResNeXt-64x4d-DCN	47.7	29.7	50.8	59.4	1.61
Det-AdvProp(NTG)	trainval35k	EfficientDet	47.6	-	-	-	-
UP-DETR[38]	trainval35k	R50	42.8	20.8	47.1	61.7	2.06
FCOS+SaAA	-	ResNeXt-101-DCN	49.6	35.7	52.5	62.4	1.39
our models							
MgD	trainval35k	EfficientNet-D3	45.6	28.1	49.8	61.1	1.62
MgD	trainval35k	EfficientNet-D5	50.0	33.5	54.4	64.1	1.49
MgD	trainval35k	EfficientNet-D7	**53.9**	**39.8**	**57.5**	**67.1**	**1.35**

all other state-of-the-art detectors but also significantly enhances the detection of small objects, achieving (AP 53.9, AP/AP_s 1.35).

5. Conclusion

In this scholarly endeavor, we discern the deleterious impact of inadequate outcomes in detecting small objects and the inherent imbalance between positive and negative samples, both of which impede the efficacy of object detectors. To redress these challenges, we proffer the introduction of Multigranular Detector (MgD), a composite framework comprising Multiscale Feature Enhancement (MFE) and Statistically Supervised Sample Weighting Strategy (S3WS) modules. Notably, the seamless integration of both MFE and S3WS modules into existing methodologies is facilitated. Experimental validations corroborate a progressive enhancement in the detectors' performance upon the incorporation of MFE and S3WS, achieved at a reasonable computational cost. Remarkably, MgD outshines all extant state-of-the-art detectors.

MgD substantiates its efficacy in ameliorating the detection of small objects, a facet underscored by the superior performance of SwinV2-G equipped with MFE and SW3S, showcasing notable improvements in average precision (AP 63.1→64.0, AP/AP_s 1.97→1.42) compared to its counterparts. MgD, exhibiting an impressive AP of 53.9 and AP/AP_s of 1.35, distinctly excels in enhancing the discernibility of small objects.

However, the innovative contributions of this study are not without discernible limitations. The MFE module, while proficient in leveraging statistical information from independent datasets, exhibits a conspicuous lack of generalization capability. Its inflexibility in adapting to diverse task scenarios becomes apparent during task transitions. The S3WS module, while constructed with basic Intersection over Union (IoU) functions, falls short in optimizing runtime efficiency and memory utilization for each IoU function. Moreover, the efficacy of SW3S is contingent upon the collective performance of several distinct IoUs.

Future endeavors will pivot towards addressing the practical deployment of the MFE module in object detection. Additionally, a concerted focus will be directed towards the nuanced intricacies of multi-scale object detection, necessitating bespoke strategies for diverse detection scenarios in computer vision. Despite the effective mitigation of sample imbalance achieved by the S3WS module, efforts will be directed towards optimizing the computational cost and memory footprint of each IoU function within SW3S. The overarching goal is to propel our work into a foundational role, fostering an evaluative feedback mechanism within computer vision subtasks characterized by expedited evaluations and streamlined model dimensions.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China 61976158 and Grant Nos. 62006172.

References

[1] Y. Ji, H. Zhang, Z. Zhang, M. Liu, Cnn-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances, Information Sciences 546 (2021) 835–857.

[2] Y. Liu, P. Sun, N. Wergeles, Y. Shang, A survey and performance evaluation of deep learning methods for small object detection, Expert Systems with Applications 172 (2021) 114602.

[3] M. Bello, G. Nápoles, L. Concepción, R. Bello, P. Mesejo, Ó. Cordón, Reprot: Explaining the predictions of complex deep learning architectures for object detection through reducts of an image, Information Sciences 654 (2024) 119851.

[4] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, J. Hemanth, Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2, Sustainable cities and society 66 (2021) 102692.

[5] L. Zhu, F. Lee, J. Cai, H. Yu, Q. Chen, An improved feature pyramid network for object detection, Neurocomputing 483 (2022) 127–139.

[6] N. Sambyal, P. Saini, R. Syal, V. Gupta, Aggregated residual transformation network for multistage classification in diabetic retinopathy, International Journal of Imaging Systems and Technology 31 (2) (2021) 741–752.

[7] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of yolo algorithm developments, Procedia Computer Science 199 (2022) 1066–1073.

[8] J. Chen, D. Liu, T. Xu, S. Wu, Y. Cheng, E. Chen, Is heuristic sampling necessary in training deep object detectors?, IEEE Transactions on Image Processing 30 (2021) 8454–8467.

[9] B. Q. Hu, Three-way decisions space and three-way decisions, Information sciences 281 (2014) 21–52.

[10] X. Yang, T. Li, H. Fujita, D. Liu, Y. Yao, A unified model of sequential three-way decisions and multilevel incremental processing, Knowledge-Based Systems 134 (2017) 172–188.

[11] Z. Wu, C. Shen, A. Van Den Hengel, Wider or deeper: Revisiting the resnet model for visual recognition, Pattern Recognition 90 (2019) 119–133.

[12] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, L. Zhang, Dynamic head: Unifying object detection heads with attentions, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 7373–7382.

[13] 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.

[14] Y. Zhang, M. Chi, Mask-r-fcn: A deep fusion network for semantic segmentation, IEEE Access 8 (2020) 155753–155765.

[15] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2020.

[16] R. Yang, Y. Yu, Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis, Frontiers in Oncology 11 (2021) 573.

[17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis & Machine Intelligence 39 (6) (2017) 1137–1149.

[18] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, K. H. Abdulkareem, Real-time hand gesture recognition based on deep learning yolov3 model, Applied Sciences 11 (9) (2021) 4164.

[19] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, R. Wang, Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection, Information Sciences 522 (2020) 241–258.

[20] D. Yang, Y. Zhou, A. Zhang, X. Sun, D. Wu, W. Wang, Q. Ye, Multi-view correlation distillation for incremental object detection, Pattern Recognition 131 (2022) 108863.

[21] L. Wei, G. Zong, Ega-net: Edge feature enhancement and global information attention network for rgb-d salient object detection, Information Sciences 626 (2023) 223–248.

[22] H. Zhang, X.-g. Hong, L. Zhu, Detecting small objects in thermal images using single-shot detector, Automatic Control and Computer Sciences 55 (2) (2021) 202–211.

[23] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[24] Y. Ji, H. Zhang, F. Gao, H. Sun, H. Wei, N. Wang, B. Yang, Lgcnet: A local-to-global context-aware feature augmentation network for salient object detection, Information Sciences 584 (2022) 399–416.

[25] H. Wang, Q. Wang, P. Li, W. Zuo, Multi-scale structural kernel representation for object detection, Pattern Recognition 110 (2021) 107593.

[26] C. Yang, Z. Huang, N. Wang, Querydet: Cascaded sparse query for accelerating high-resolution small object detection, in: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, 2022, pp. 13668–13677.

[27] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, J. Han, Towards large-scale small object detection: Survey and benchmarks, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[28] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6054–6063.

[29] A. Luque, A. Carrasco, A. Martín, A. de Las Heras, The impact of class imbalance in classification performance metrics based on the binary confusion matrix, Pattern Recognition 91 (2019) 216–231.

[30] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 761–769.

[31] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007.

[32] B. Li, Y. Liu, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8577–8584.

[33] K. Oksuz, B. C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: A review, IEEE transactions on pattern analysis and machine intelligence 43 (10) (2020) 3388–3415.

[34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision(ICCV), 2017, pp. 764–773.

[35] S. Zhang, C. Chi, Y. Yao, Z. Lei, S. Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in: CVPR, 2020.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv preprint arXiv:2103.14030.

[37] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al., Swin transformer v2: Scaling up capacity and resolution, arXiv preprint arXiv:2111.09883.

[38] Z. Dai, B. Cai, Y. Lin, J. Chen, Up-detr: Unsupervised pre-training for object detection with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1601–1610.

[39] D. Du, L. Wen, P. Zhu, H. Fan, Z. Liu, Visdrone-det2020: The vision meets drone object detection in image challenge results, IEEE.

[40] Y. Cao, Z. He, L. Wang, W. Wang, Y. Yuan, D. Zhang, J. Zhang, P. Zhu, L. Van Gool, J. Han, et al., Visdrone-det2021: The vision meets drone object detection challenge results, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2847–2854.

[41] L. A. Ibrahim, S. Huang, M. Fernandez-Otero, M. Sherer, Y. Qiu, S. Vemuri, Q. Xu, R. Machold, G. Pouchelon, B. Rudy, et al., Bottom-up inputs are required for establishment of top-down connectivity onto cortical layer 1 neurogliaform cells, Neuron 109 (21) (2021) 3473–3485.

[42] K. Pang, D. Ai, H. Fang, J. Fan, H. Song, J. Yang, Stenosis-detnet: Sequence consistency-based stenosis detection for x-ray coronary angiography, Computerized Medical Imaging and Graphics 89 (2021) 101900.

[43] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, Sod-mtgan: Small object detection via multi-task generative adversarial network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 206–221.

[44] P. Zhou, B. Ni, C. Geng, J. Hu, Y. Xu, Scale-transferrable object detection, in: proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 528–537.

[45] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, A. L. Yuille, Single-shot object detection with enriched semantics, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5813–5821.

[46] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 385–400.

[47] J. Wang, K. Chen, S. Yang, C. C. Loy, D. Lin, Region proposal by guided anchoring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2965–2974.

[48] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, J. Shi, Foveabox: Beyound anchor-based object detection, IEEE Transactions on Image Processing 29 (2020) 7389–7398.

[49] C. Zhu, Y. He, M. Savvides, Feature selective anchor-free module for single-shot object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 840–849.

[50] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[51] X. Zhou, J. Zhuo, P. Krahenbuhl, Bottom-up object detection by grouping extreme and center points, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 850–859.

[52] K. Takeuchi, I. Yanokura, Y. Kakiuchi, K. Okada, M. Inaba, Automatic learning system for object function points from random shape generation and physical validation, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 2428–2435.

[53] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6569–6578.

[54] Z. Yang, S. Liu, H. Hu, L. Wang, S. Lin, Reppoints: Point set representation for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9657–9666.