Multi-granularity Detector Focusing on Size-different Objects and Positive and Negative Samples Imbalance

Chen Dong

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

Miao Duoqian[1,*]

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

Zhao Xuerong

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

Abstract

This paper revisits object detection models and points out that the performance of detectors is restricted by poor results of small objects and imbalance between positive and negative samples. To those ends, we propose **M**ulti-**g**ranularity **D**etector (MgD), in which the main ingredients are **M**ulti-granularity **F**eature **E**xtraction (MFE) and **S**equential three-**w**ay **S**election (S3WS). In MFE, depending on the analysis of different-size objects, we apply three multi-granularity customizable deformable convolutions to three layers of feature maps. MFE improves the results of small objects, which in turn improves the performance of general object detection. Meanwhile, we propose S3WS to ameliorate the imbalance between positive and negative samples. Region proposals are fed into S3WS, then more positive samples are selected from positive and boundary regions according to multiple evaluation functions and two dynamical thresholds layer by layer. Extensive experiments on COCO benchmark prove that

[*]Corresponding author
[1]Deputy director of Key Laboratory of embedded system and service computing, Ministry of Education

MgD outperforms other state-of-the-art models in system level. Meanwhile, 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. MgD [1] (AP 53.9, AP/AP_s 1.35) greatly improves the contribution of small objects. Moreover, MFE and S3WS can be easily integrated into ConvNet detectors and transformer-based detectors, and achieve significant improvements.
Keywords: Computer Vision, Deep Learning, Object Detection, Granular Computing

1. Introduction

The object detection aims to find objects and to determine their classes and locations in an image. Convolutional neural networks (ConvNets) provide a strong impetus to the field of object detection. With the help of increasingly large neural networks and progressively more complex convolution structures, the detection results have seen significant improvement in recent years. However, scholars have focused on the designing of neural networks and the tuning of convolution structures, leading to poor performance for small objects comparing to that for medium and large objects. Ultimately, this results in a less progress of general object detection.

Generally speaking, the performance for small objects is limited, which affects the weak growth of performance for all objects. Table 1 shows that AP_s lags far behind AP, AP_m, and AP_l. Improving the performance of small objects [1] will achieve a significant progress for general object detection.

The early explicit attempts to improve the incongruity of different-size object detection are SSD [2] and FPN [3]. The single-granularity vanilla convolution kernel (generally 3×3 or 5×5 [4, 5]) will limit feature extraction of different-size objects in *backbone*. However, the analysis of the dataset and the consideration for the characteristics of different-size objects are still overlooked. It should be focused more on the analysis of datasets and kernel sizes for different-size objects in *backbone*. To this

[1] The 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**

end, we propose the **M**ulti-**g**ranularity **D**etector (MgD), in which the main ingredients are **M**ulti-granularity **F**eature **E**xtraction (MFE, or *stomach*) and **S**equential three-**w**ay **S**election (S3WS). The MgD is based on a reconstruction of network architectures and redesign of evaluation functions at the surgical level.

To improve the incongruity of different-size object detection, the MFE module consists of multi-granularity deformable convolution kernels which are customizable for different-size objects. The kernel for small objects is customized based on ones in backbone at first. Then, the scale factors k_1 and k_2 are determined, which in turn will determine the size of the deformable convolution kernel for medium and large objects. Three customizable deformable convolutions are applied to three feature maps released from *backbone*. Each feature map together with its customizable deformable convolutions forms a stomach net, and three stomach nets form a *stomach* module.

Furthermore, the existing object detection models suffer from the problem of imbalance between positive and negative samples, which also affects the final performance.

The detectors will get a significant progress, if the rate of positive and negative samples is close to 1:3[6]. We propose S3WS to ameliorate this problem. Region proposals generated by the neural network are scored by multiple evaluation functions, which are fed to S3WS module. The region proposals x with an evaluation value $IoU_i(x)$ greater than α_i belongs to the set of positive samples, less than β_i belongs to the set of negative samples, and in between α_i and β_i belongs to the set of the boundary region[7, 8]. Besides, the region proposals in the set of boundary region enter into the next level of classification until the stopping criterion is reached. Positive samples are selected from positive and boundary regions according to multiple evaluation functions and two dynamical thresholds layer by layer, but negative samples are selected in one layer. The two thresholds α_i and β_i are dynamically determined by the evaluation function and the region proposals of the same batch size.

The main contributions of this work are summarized as:

- We propose that the detection result of small objects and imbalance between positive and negative samples restrict the general detector performance.

- We propose that MFE module consists of multi-granularity deformable convolution kernels to improve the incongruity of different-size object detection. Meanwhile, S3WS is proposed to ameliorate this problem of imbalance between positive and negative samples.

- To those ends, we propose MFE and SW3S modules,which can be easily integrated into ConvNet detectors [9] and transformer-based detectors [10] and achieve significant improvements. Meanwhile, 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 with slightly bigger model size.

- Our method MgD(see Table10) outperforms all other state-of-the-art ones on MS COCO[11] and improves the contribution of small objects.

2. Related Work

Our work build on prior ones in several domains: analysis of datasets, rethinking on backbone architectures and convolution kernels, and redesigning of evaluation functions.

2.1. CNN and variant

The R-CNN series and YOLO series are the classical representatives of two-stage model [12] and one-stage model [13], respectively, in object detection. The first culmination of deep learning for object detection was R-CNN [14]. The Fast R-CNN [14] and Faster R-CNN [15] (show in Figure 1) models are the basic framework for deep learning applying to object detection. YOLO provides a more straightforward way by directly regressing the location of the bounding box and determining the class to which the bounding exploitation belongs, thus it transforms the object detection problem into a regression problem. Afterwards, various YOLO models [5, 16] were proposed, which improves not only the accuracy but also the computing speed of the deep learning network. However, these models ignore the difference between large and small objects in the dataset.

2.2. Backbone architectures

The SSD [2] and FPN [3] 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 [17].

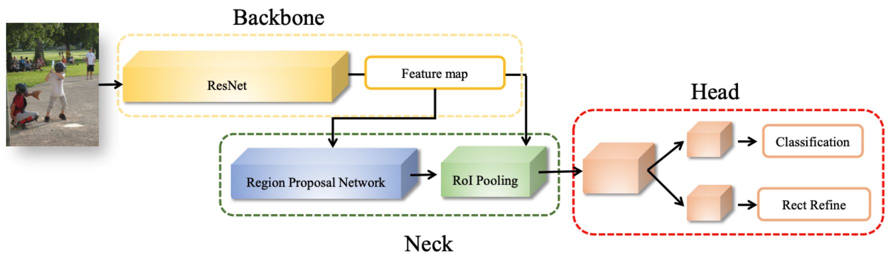

Figure 1: The main components of traditional object detection model. Most existing detection models mainly consist of *backbone*, *neck* and *head*. Note that we only show Faster R-CNN as an example.

2.3. Convolution kernels

In the meanwhile, the convolution kernel[18] evolves. For example, deformable conv [19] adds an offset variable to the position of each sampled point in the convolution kernel, enabling random sampling around the current position without being restricted to the previous regular grid points; dilated convolution focuses on the semantic information of the local pixel blocks by letting each pixel aggregate with the surrounding blocks, which affects the detail of segmentation[20].

2.4. Imbalance between positive and negative samples

Larger neural networks and more complex convolutional structures constantly aggravate the problem of positive and negative sample imbalance [21]. In machine learning, one can solve the sample imbalance problem from the data and the algorithm perspective. For example, data enhancement, OHEM, and GHM.

3. The MgD model

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.

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

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

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

$$y(k)=\sum_{j=1}^{n_k} x\left(p_{kj}+\Delta p_k\right) \cdot \Delta m_k / n_k, \tag{2}$$

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], while Δp_k can be any value. And p_{kj} is the sampling location for the j th grid cell in the k th bin, and n_k denotes the number of sampled grid cells.

The following formulas are used to determine the value of k_1 and k_2 :

$$\frac{KS_{\text{small}}}{KS_{\text{medium}}}=\sqrt{\frac{Aa_{\text{small}}}{Aa_{\text{medium}}}}=\frac{1}{k_1} \tag{3}$$

$$\frac{KS_{\text{medium}}}{KS_{\text{large}}}=\sqrt{\frac{Aa_{\text{medium}}}{Aa_{\text{large}}}}=\frac{1}{k_2} \tag{4}$$

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 convolutional 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

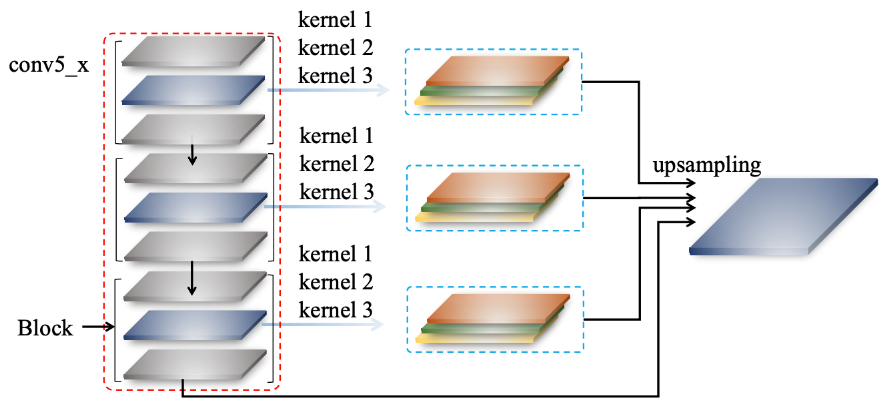

Figure 2: *Stomach net*. Here, we adopt ResNet101 as an example. In con5_x layer of ResNet101, every feature map after kernel 3×3 (blue ones in three blocks) are utilized with multi-granularity kernels (kernel 1, kernel 2, and kernel 3).

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.

3.3. Sequential three-way selection for region proposals

The existing detectors suffer from severe imbalance between positive and negative samples. In this section, we propose a S3WS module, which combines the idea of sequential three-way decision with selection module, to ameliorate the imbalance of positive and negative samples. A sequential three-way decision consists of a series of three-way decision. The key idea of three-way decision is to divide a set of objects into positive, negative, and boundary regions based on evaluation functions and decision parameters α and β. The objects in the positive and negative regions are with certain decisions, namely, acceptance and rejection. For objects in boundary region, another

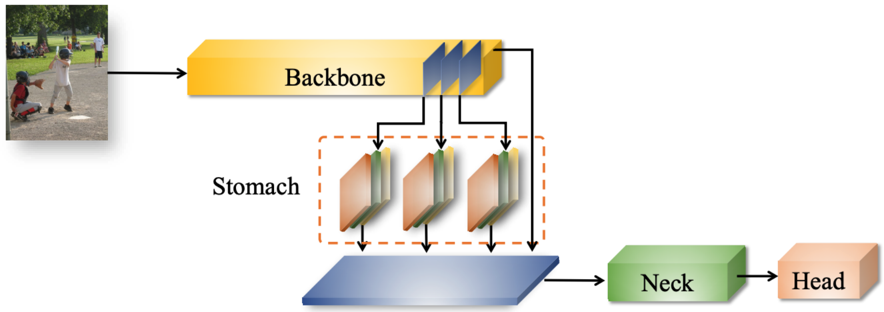

Figure 3: *Stomach*.

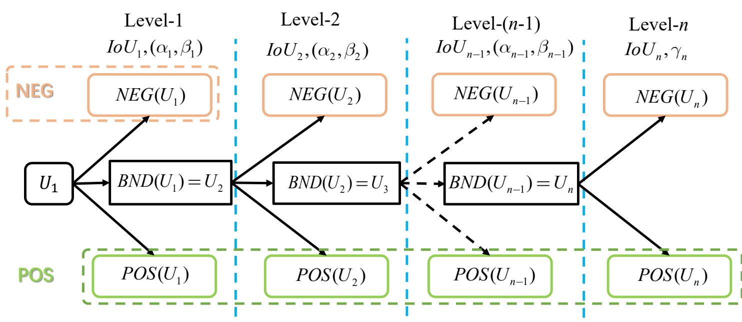

Figure 4: Sequential three-way selection module. POS means positive sample set and NEG means negative sample set. $POS = POS\,(U_1) \cup \cdots \cup POS\,(U_n)$ and $NEG = NEG\,(U_1)$.

process of three-way decision is conducted [7, 8].

145 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 $IoU_i = tIoU^2, GIoU, CIoU, DIoU, or$ $CDIoU$. 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:

[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.

$$
\begin{aligned}
&\alpha_i = \frac{1}{m}\sum_{j=1}^{m} IoU_i(x) \\
&\beta_i = \alpha_i - \sqrt{\frac{\sum_{j=1}^{m}\left(IoU_i(x) - \alpha_i\right)^2}{m}} \\
&i = 1, 2, ..., n-1
\end{aligned} \tag{5}
$$

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):

$$
\begin{aligned}
&\mathrm{POS}\,(U_1) = \{x \in U_1 \mid IoU_1(x) \geq \alpha_1\} \\
&\mathrm{BND}\,(U_1) = \{x \in U_1 \mid \beta_1 < IoU_1(x) < \alpha_1\} \\
&\mathrm{NEG}\,(U_1) = \{x \in U_1 \mid IoU_1(x) \leq \beta_1\}
\end{aligned} \tag{6}
$$

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:

$$
\begin{aligned}
&\mathrm{POS}\,(U_2) = \{x \in U_2 \mid IoU_2(x) \geq \alpha_2\} \\
&\mathrm{BND}\,(U_2) = \{x \in U_2 \mid \beta_2 < IoU_2(x) < \alpha_2\} \\
&\mathrm{NEG}\,(U_2) = \{x \in U_2 \mid IoU_2(x) \leq \beta_2\}
\end{aligned} \tag{7}
$$

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 :

$$\begin{aligned} &\operatorname{POS}\left(U_{n-1}\right)=\left\{x \in U_{n-1} \mid IoU_n(x) \geq \gamma_n\right\} \\ &\operatorname{NEG}\left(U_{n-1}\right)=\left\{x \in U_{n-1} \mid IoU_n(x)<\gamma_n\right\} \end{aligned} \tag{8}$$

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, but the regression loss function of whole detector will be expressed as:

$$\mathcal{L}_{\text{reg}}=\mathcal{L}_{\mathrm{IoU}_1}+\mathcal{L}_{\mathrm{IoU}_2}+\cdots+\mathcal{L}_{\mathrm{IoU}_n}. \tag{9}$$

4. Experiments

In this section, we first introduce the datasets and hardware information. Then, we describe the implementation details of the experiment, including ablation studies on MFE and S3WS modules. Finally, we compare our method with the state-of-the-art ones.

Settings. The following experiments were conducted on MS COCO 2017 and PASCAL VOC 2012 dataset using two GeForce RTX 3090 GPUs and two Tesla V100 PCIe 32GB GPUs. All models under pytorch or tenserflow framework are standard models without using any tricks. For the ablation study and comparisons, we consider four typical object detection frameworks: ATSS[22], Faster RCNN[15], Swin Transformer (V1, V2)[23, 24], and DETRs (DETR, UP-DETR)[25].

Dataset. We perform experiments on COCO 2017 detection datasets, containing 118k training images, 5k validation images and 20K test-dev images. The ablation study is performed using the validation set, and a system-level comparison is reported on test-dev. Each image is annotated with bounding boxes and panoptic segmentation. There are 7 instances per image on average, up to 63 instances in a single image in training set, ranging from small to large on the same images.

PASCAL VOC 2012 provided a total of 17125 pictures of different sizes, covering four categories of people, animals, vehicles, and indoor furniture, as well as its sub categories, totaling 20 categories of pictures. The training data provided consists of a set of images; each image has an annotation file giving a bounding box and object class label for each object in one of the twenty classes present in the image. Note that multiple objects from multiple classes may be present in the same image. Annotation was performed according to a set of guidelines distributed to all annotators. The data has been split into 50% for training/validation and 50% for testing. The distributions of images and objects by class are approximately equal across the training/validation and test sets.

textbfTraining. MgD is trained with Adamw and SGD optimizers, changing Adamw to SGD until very final stage. We adopt MgD model with EfficientNetD3/D5/D7 backbone and the learning rate for backbone is 2^{-5}. We follow the DETR training protocol. The backbone is the ImageNet-pretrained model from TORCHVISION with batchnorm layers fixed, and the transformer parameters are initialized using the Xavier initialization scheme. The weight decay is set to be 10^{-4}. The settings of ATSS and Faster RCNN are in accordance with [22]. During training, we apply horizontal flipping and scale jittering [0.1, 2.0], which randomly rsizes images between 0.1x and 2.0x of the original size before cropping. We apply soft-NMS for eval.

We evaluate MgD on COCO 2017 detection datasets with 118K training images. Each model is trained using SGD optimizer with momentum 0.9 and weight decay 4^{-5}. Learning rate is linearly increased from 0 to 0.16 in the first training epoch and then annealed down using cosine decay rule. Synchronized batch norm is added after every convolution with batch norm decay 0.99 and epsilon 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

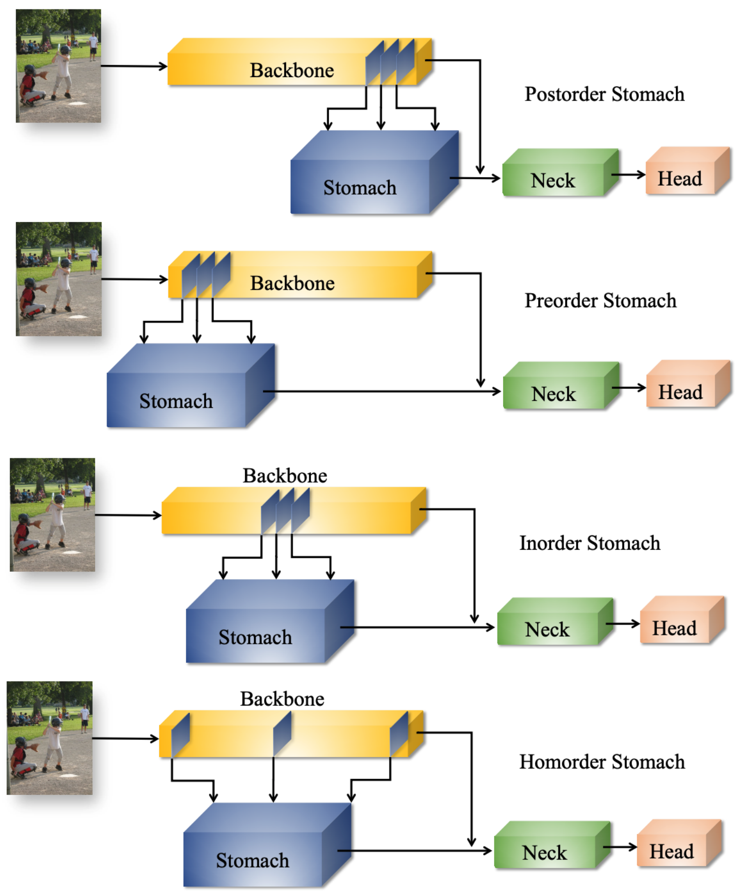

Figure 5: *Stomaches* in different positions. *Postorder stomach*: *stomach* in the last three layers of *backbone*; *inorder stomach*: in the middle three layers; *preorder stomach*: in the first three layers; *homorder stomach*: *stomach nets* are homogeneously distributed in *backbone*.

the front of *backbone*, the detection results decrease rapidly and reach the lowest for *homorder stomach*.

210 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:

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

- According to Formula 3 and 4, 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.

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

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$\sim$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

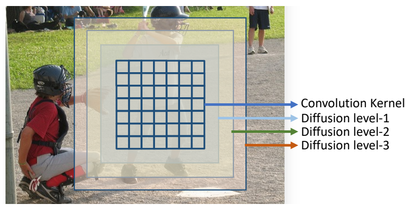

Figure 6: Deformable convolution kernel with different diffusion. The blue grids represent the deformable convolution kernel. The blue translucent rectangular box symbolizes the convolution kernel diffusing outward by one pixel unit; the green one symbolizes two pixel units; the red one symbolizes three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

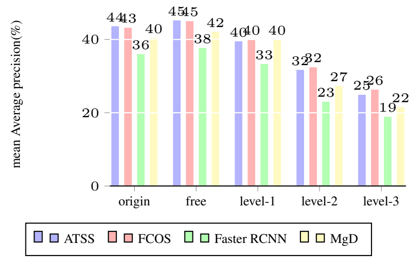

Figure 7: Detection results (%) with different *diffusion levels* on MS COCO *validation* set. Level-1 = *diffusion level-1*; Level-2 = *diffusion level-2*; Level-3 = *diffusion level-3*; Free = *free diffusion*.

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:

- 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

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

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. Comparison

Through the above ablation studies, we have obtained the best configuration of MFE and S3WS. In order to verify the effectiveness of MFE and S3WS, we also designed the following comparative experiments on representative models. Results are presented in Table 7. The performance of detectors gradually gets significant improvements by adding MFE and S3WS.

From Table 9, 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.

Comparison to traditional ConvNets. The performance of ATSS and Faster RCNN with/without MFE and SW3S is shown in Table 8 left part. The results with MFE and SW3S are +1.1/+2.0 AP and -0.36/-0.55 AP/AP_s higher/lower than those without them. Model size and inference speed have hardly changed.

Table 7: 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[25] 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 8: 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 8 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 9: Detection results (%), FPS, model size, and testing time on MS COCO *validation* set.

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

Comparison to transformer-based methods. Table 8 right part shows that the results of Swin Transformer, Swin Transformer V2, DETR, and UP-DETR with/without MFE and SW3S. It is obvious that transformer-based methods suffer from poor result of small object and imbalance between positive and negative samples. Then MFE and SW3S achieve significant improvements. The results with MFE and SW3S are +1.2/+1.3 AP and -0.47/-0.49 AP/AP_s higher/lower than those without them. Meanwhile, 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.

Comparison to previous state-of-the-art. A new detector MgD is designed by adding MFE and S3WS modules after the EfficientNet. Comparing MgD with other object detectors, we found that MgD outperforms all other state-of-the-art ones on MS COCO dataset in Table 10. Meanwhile, MgD(AP 53.9, AP/AP_s 1.35) greatly improves the contribution of small objects.

Table 10: 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[12]	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[26]	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[27]	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[28]	trainval35k	RerNet-101	41.4	24.7	44.2	52.6	1.68
anchor-based one-stage							
YOLOv2[5]	trainval35k	DarkNet-19	21.6	5.0	22.4	35.5	4.32
SSD512[2]	trainval35k	VGG-16	28.8	10.9	31.8	43.5	2.64
STDN513[29]	trainval	DenseNet-169	31.8	14.4	36.1	43.4	2.21
DES512[30]	trainval35k	VGG-16	32.8	13.9	36.2	47.5	2.36
DSSD513[18]	trainval35k	ResNet-101	33.2	13.0	35.4	51.1	2.55
RFB512-E[31]	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[32]	trainval35k	ResNet-50	39.8	21.8	42.6	50.7	1.83
FoveaBox[33]	trainval35k	ResNeXt-101	42.1	24.9	46.8	55.6	1.69
FSAF[34]	trainval35k	ResNeXt-64x4d-101	42.9	26.6	46.2	52.7	1.61
FCOS[35]	trainval35k	ResNeXt-64x4d-101	43.2	26.5	46.2	53.3	1.63
anchor-free keypoint-based							
ExtremeNet[36]	trainval35k	Hourglass-104	40.2	20.4	43.2	53.1	1.97
CenterNet-HG[37]	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[38]	trainval35k	Hourglass-104	44.9	25.6	47.4	57.4	1.75
RepPoints[39]	trainval35k	ResNeXt-101-DCN	45.0	26.6	48.6	57.5	1.69
recent excellent models							
ATSS[22]	trainval35k	ResNeXt-64x4d-DCN	47.7	29.7	50.8	59.4	1.61
Det-AdvProp(NTG)	trainval35k	EfficientDet	47.6	-	-	-	-
UP-DETR[25]	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**

5. Conclusion

In this work, we identify that poor results of small objects and imbalance between positive and negative samples restrict the performance of detectors. To address these issues, MgD are proposed, which consists of MFE and S3WS modules. Both MFE and S3WS modules can be integrated into the existing methods easily. The experiments demonstrate that the performance of detectors gradually gets significant improvements by adding MFE and S3WS at an acceptable cost. Furthermore, the MgD detector outperforms all other state-of-the-art ones. The MgD **does** improve the contribution of small objects. Meanwhile, 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. MgD(AP 53.9, AP/AP_s 1.35) greatly improves the contribution of small objects.

But the innovation of this paper also has obvious limitations. The MFE module is mainly limited to the statistical information of independent data sets, and obviously lacks the generalization ability. When switching task scenarios, the MFE module lacks flexibility. The S3WS module is stacked by basic IoU functions, and does not compress the running time and memory space of each IoU function. At the same time, the performance of SW3S is subject to the combination of the performance of several different IoUs.

In the future work, we will mainly solve the application of MFE module in object detection. At the same time, we should pay attention to size-different objects customarily. Size-different objects should use different detection strategies. Customized solutions should be adopted for various objects in computer vision in the future. Although the S3WS module effectively alleviates the imbalance between positive and negative samples, it does not compress the running time and memory space of each IoU function. At the same time, the performance of SW3S is subject to the combination of the performance of several different IoUs. In the future, we will mainly solve the problem of operating cost. We do hope that our work will play a role of cornerstone to encourage the evaluation-feedback mechanism in computer vision subtasks with less time and lighter model size.

Acknowledgements

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

References

[1] 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.

[2] 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.

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

[4] 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.

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

[6] 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.

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

[8] 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.

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

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.