YOLOAX: YOLOX With Attention

Kejian Xu
School of Computer Science and Information Security
Guilin University of Electronic Technology
Guilin, China
xkj897899031@gmail.com

*Jinlong Chen
School of Computer Science and Information Security
Guilin University of Electronic Technology
Guilin, China
Jinlong.chen@guet.edu.cn

MingHao Yang
Institute of Automation
Chinese Academy of Sciences
Beijing, China
21032303135@mails.guet.edu.cn

Yi Ning
School of Continuing Education
Guilin University of Electronic Technology
Guilin, China
296106092@qq.com

abstract—Real-time object detection is always a vital topic in computer vision. To achieve a great accuracy-speed trade-off of object detectors has always been an extremely challenging task for academia and industry. Although recent transformer-based models have demonstrated the advantage of attention mechanism and achieved impressive performance boost over CNNs, the computational overload may harm the performance when taking real-time object detection tasks. YOLO series has been playing an irreplaceable role in object detection. In this paper, we opt for YOLOX as our strong baseline and present some effective improvements to YOLOX, forming a new high-performance detector YOLOAX. For further enjoying the benefit of attention mechanism, we propose some attention-based modules that can activate CNNs to highlight the most salient regions and enhance the ability to learn the most informative image representations of the feature maps. Furthermore, we introduce a new leading label assignment strategy STA and a new loss function GEIOU Loss to optimize our detector. We verify the effectiveness of our proposed methods through extensive ablation studies on COCO and VOC 2012 detection datasets. Finally, we train YOLOAX only on COCO from scratch without any pre-trained weights, and get 54.2% AP on the COCO 2017 test set at a real-time speed of 72.3 FPS, outperforming YOLOX by 2.7% AP. Source code is released at https://github.com/KejianXu/yoloax.

Keywords: **Real-time object detection; Attention mechanism; Region boosting; Label assignment strategy; Loss function**

I. INTRODUCTION

Object detection, especially real-time object detection, is always a very important challenging topic in computer vision systems, requiring the detector to predict a bounding box with a category label as accurately and quickly as possible for each instance of interest in an image. Driven by the success of region-based convolutional neural networks (R-CNNs) [9] and region proposal methods [18], the later incarnations such as Fast R-CNN, Faster R-CNN, Mask R-CNN and others [13, 14, 15, 44] pursuit higher accuracy while ignoring the time spent on region proposals and high latency, which show poor performance for taking real-time object detection tasks. With the development of object detection, research attention has been geared towards one-stage object detection. YOLO series [27, 28, 11, 5, 4, 12, 1, 2] innovatively choose to trade in much faster computing speed at the cost of appropriately reduced accuracy, which achieve a better trade-off between high accuracy and optimal speed for real-time object detection tasks. Recently with the introduction of the anchor-free manner [29, 9], the major advances with anchor-free in object detection academia like YOLOX [1] and YOLOR [12] have achieved significant performance boost over previous efforts, which also focus on applying new advanced label assignment strategies [31, 34, 37, 38] to automatically classify positive and negative training samples, dramatically increasing speed with guaranteed accuracy, instead of using hand-crafted assigning rules for training.

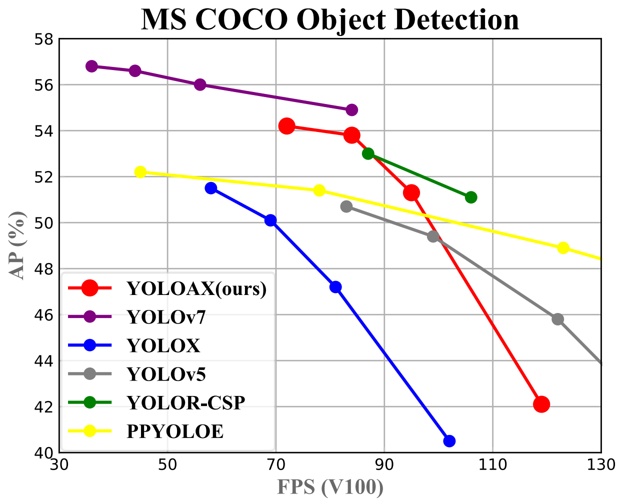


Figure 1. Comparison of our proposed YOLOAX and other real-time object detectors, our methods achieve the best accuracy-speed trade-off.

At the same time these recent years, as for Vaswani et al. (2017) [19] proposing Transformer, a simple network architecture based exclusively on multi-headed attention mechanisms for sequence transduction tasks, which holds the best trade-off performance at that time. The follow-up efforts [40, 55, 23, 32, 47] attempt to apply the Transformer on multiple vision tasks and begin to explore the attention mechanism in depth, which all attain state-of-the-art results on various vision tasks. The Transformer-based models not only significantly improve the speed, but also reduce additional computational cost, which demonstrate the advantages of the attention. Despite the impressive achievements, those super large-scale models suffer from poor performance on the tasks in real-time object detection. As for conducting real-time object detection tasks, we experientially argue that conventional attention-based approaches still have two main limitations: (1) Requiring enormous computing resources if the model bases solely on attention; (2) Especially relying on strong data augmentation strategies or adequate annotations.

That's what brings us here. Recent efforts [33, 54, 56] show the benefit of combining transformer and convolution in series, regardless of the part in which the convolution is used. Most current mainstream object detectors are developed for GPU and use ResNet [10], CSPNet [20], or DarkNet [11] as the backbone of the network architecture with different methods and strategies to optimize. In this paper, considering the latest model may be a little over-optimized for the anchor-free pipeline, we choose YOLOX with CSPDarknet as our baseline and deliver the aforementioned advancements to YOLOX with experienced optimization. As shown in Figure 1, absorbing the essence of attention mechanism and equipped with the effective methods we proposed, we boost the YOLOX to 47.3% AP (YOLOAX) on COCO with 640×640 resolution, exceeding the counterpart YOLOX by 1.8% AP with fewer additional computational cost. We also conduct extensive ablation experiments to verify the influence of our design methods during the detector training. We have released our source code at https://github.com/KejianXu/yoloax.

Our main contributions are summarized as follows:

- We propose two attention-based modules, the cross-stage partial module and the spatial feature pooling module, which can significantly highlight the region of interest and boost representation power of CNNs.
- We introduce a new advanced label assignment (STA) to optimize the assigning procedure of object detectors that bring faster computing speed and fewer additional inference cost.
- We modify the YOLO's head and present the Generalized Efficient IOU Loss that can effectively reduce the regression loss of object detectors.

II. RELATED WORK

Below, various key approaches that are most relevant to our work will be discussed.

A. One-stage object detectors

A conventional detector usually consists of two parts, a backbone for extracting effective features, which is pre-trained on your dataset, and a head for predicting classes and regression of bounding boxes of objects. In recent years, the most advanced two-stage object detectors are mostly based on the R-CNN series [9, 13, 14, 15], which focus on locating the object to get multi-scale anchors to ensure sufficient accuracy and recall in the first stage, and then classify the anchors to find a more precise location in the second stage. Nevertheless, as the development of one-stage object detectors, they are slow and hard to optimize comparing to state-of-the-art one-stage object detectors. The most representative one-stage object detectors are mainly based on YOLO series [27, 28, 11, 5, 4, 12, 2], SSD [24] and RetinaNet [8], which can achieve much faster detection speeds at the cost of slightly reduced accuracy. To get faster speed without increasing additional inference cost, the one-stage object detectors with anchor-free manner are developed. The models of this sort are CornerNet [35], CenterNet [53], FCOS series [21, 39], etc. Currently, the most representative real-time object detectors usually require a stronger and faster network architecture as the backbone like ResNet, CSPNet, DarkNet, etc. and may integrate various layers or more effective methods to further optimize the model. Some researchers put their attention on designing a new whole model or a new powerful backbone for object detection, which are [36, 51, 52]. By contrast, some efforts aim to explore more effective approaches for detection. Lin et al. (2017) [18] proposed a new mechanism called Feature Pyramid Network (FPN), which can learn informative image representations by fusing feature maps collecting from different shapes. Obviously shown in previous researches [7, 8, 34, 17, 1], a more robust loss function and an advanced label assignment strategy plays an important role for models achieving high performance. One of the main contributions of this paper is designing some new advanced learning methods associated with the mentioned above for optimizing the model.

B. Attention mechanism

As for human visual system, one important property is that humans can selectively focus on salient parts through a sequence of partial glimpses when staying in a whole scene, which is largely attributable to attention mechanisms. Recently, the significance of attention in deep learning has been studied extensively in previous literature [41, 42, 43, 26]. Especially the proposal of Transformer took the research of attention to a new level. The follow-up models [40, 55, 23, 32, 47] achieved impressive performance not only on multiple computer vision tasks but also on nature language processing tasks, showing the power of attention mechanism to tell where to focus and also improve the representation of interests. To further explore the potential of attention, in recent years, Hu et al. (2019) [6] proposed SENet, which is composed of two core parts, a squeeze module which is used to model channel dependencies of feature maps and learn the corresponding information between channels, and an excitation module which is used to adaptively learn the feature of interest based on channel-wise attention and weight the representations of different channels. Then, Wang et al. (2020) [48] propose an enhanced network structure called ECANet, applying a local cross channel interaction strategy for avoiding dimensional reduction, which result in excellence performance without increasing the model's computational complexity. Additionally, the BAM and CBAM [3, 16], which infer the attention weights along both spatial and channel dimensions in turn, can be seamlessly integrated into any CNN architectures and trained end-to-end with the base CNN to perform adaptive feature optimization. Based on these studies, one of the main goals of our work is to fully utilize the role of attention mechanism. In this paper, two new modules based on attention are proposed, to activate the model to put the attention on the region of interest and capture the most important image representations based on spatial attention, which can significantly boost the ability for global image representations learning.

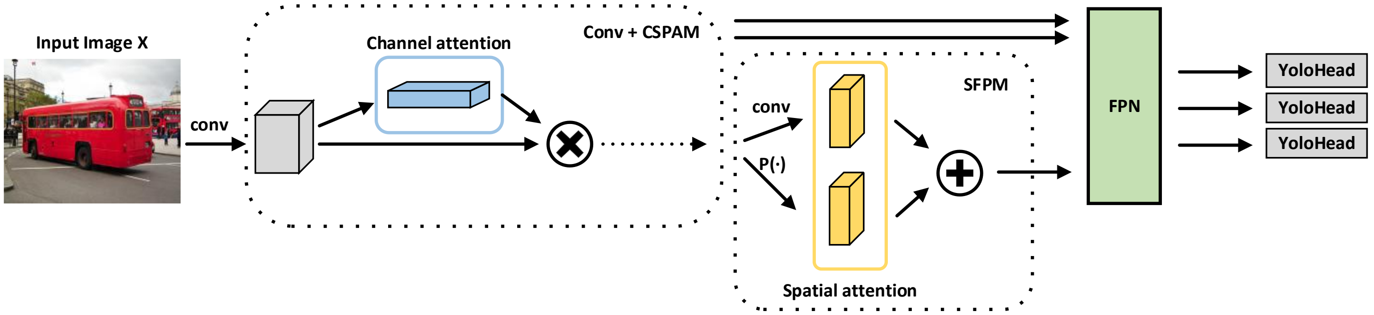


Figure 2. The overview of our proposed method. After feature extraction in CSPAMs and SFPM, the feature maps of different shapes will be fed into the FPN for feature fusion to further get the most important image representations for YoloHead to predict.

III. ARCHITECTURE

We opt for YOLOX with CSPDarknet as a strong baseline and have substantially modified the structure of YOLOX, forming a new high-performance detector YOLOAX. In the following part, we will walk through the whole architecture designs in YOLOAX step by step. The overview of the proposed YOLOAX architecture is illustrated in Figure 2.

A. *Cross-Stage Partial Attention Module*

To exploit the potential of attention mechanism, we propose a cross-stage partial attention module called CSPAM to serve as the base module of our backbone, as illustrated in Figure 4. Given a feature map $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_c] \in R^{C\times H\times W}$ as input, the CSPAM is a computational unit which can map the input X to the output $\mathrm{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \ldots, \boldsymbol{u}_c] \in R^{C\times H\times W}$. The representations of the input X will be initially learned through a channel self-attention module (CSAM) we proposed, which can activate the model to selectively inhibit less useful features and emphasize informative ones by learning to use global information. The CSAM can be embedded in any CNN modules to effectively enhance the ability of learning the representations useful for model. The whole architecture of the CSAM is shown in Figure 3 and the results of assessing the effectiveness of CSPAM are demonstrated in Table 1,2, which raises the detector from 40.5% AP to 40.8% AP.

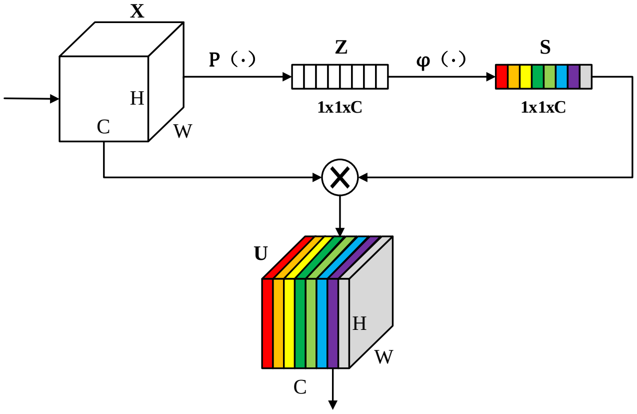


Figure 3. The architecture of the CSAM.

Considering the problem that each learned filters with a local receptive field has difficulty exploiting contextual information outside that region, we first decide to use global average pooling operation $\mathbf{P}(\cdot)$ to squeeze global spatial information into each channel descriptor and refer to each channel output as $\boldsymbol{z}_i,\ i \in [1, c]$. The statistic $\boldsymbol{Z} \in R^{C\times 1\times 1}$ is formally generated by shrinking the spatial dimensions $H \times W$ of $\boldsymbol{X}$, as defined:

$$\boldsymbol{z}_i = \mathbf{P}(\boldsymbol{x}_i) = \frac{1}{H \times W} \sum_{u=1}^{H} \sum_{v=1}^{W} \boldsymbol{x}_i(u, v) \qquad (1)$$

Subsequently, to analyze the importance of each channel and focus on the most discriminative features, we utilize 1×1 base convolutional operation $\boldsymbol{\varphi}_1(\cdot)$ twice in succession. It's our view that each value learned is compressed from the 2D features of each channel and can be thought of as having a global receptive field to some extent. The output is denoted as $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \ldots, \boldsymbol{s}_c] \in R^{C\times 1\times 1}$, which is calculated by:

$$\boldsymbol{s}_i = \boldsymbol{\varphi}_1^2(\boldsymbol{z}_i) = \boldsymbol{\varphi}_1^2(\mathbf{P}(\boldsymbol{x}_i)) \qquad (2)$$

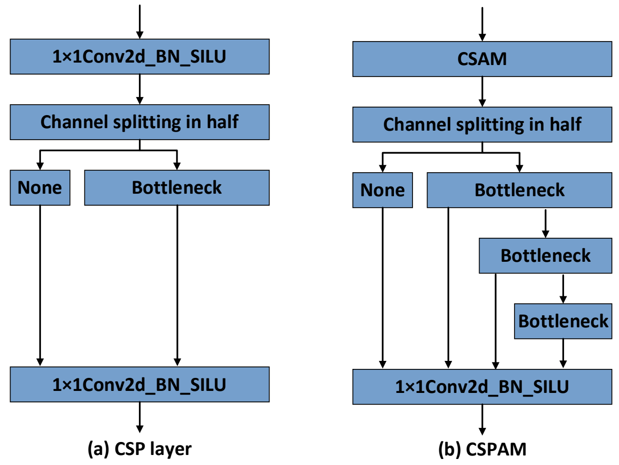
where $\boldsymbol{\varphi}_1(\cdot)$ represents a standard 2d convolution with batch normalization and SiLU activation function.

To mitigate negative activations and normalize $\boldsymbol{S}$, as for non-linear activation function, instead of ReLU, our work opt for the smoother activation function SiLU, which outperforms ReLU especially training deep learning models. By utilizing the $\boldsymbol{S}$ as normalized weighting factors or scalar, the two dimensions of height and width of the input $\boldsymbol{X}$ are rescaled with $\boldsymbol{S}$ in each channel to obtain the final feature map $\boldsymbol{U} \in R^{C\times H\times W}$, which adaptively highlight the more informative representation of the feature map:

$$\boldsymbol{U} = \boldsymbol{S} \otimes \boldsymbol{X} \qquad (3)$$

$\otimes$ represents channel-wise multiplication.

Since the outstanding performance of residual learning framework, our proposed CSPAM adopts the bottleneck architecture of the CSPDarknet to mitigate the gradient disappearance, and optimize the network degradation problems and the training difficulties. For the main considerations that reducing the computational density and the number of hyperparameters, the number of channels of the input $\boldsymbol{U}$ will first be expanded to twice its original size after a 1×1 base convolutional operation $\boldsymbol{\varphi}_1(\cdot)$, and then the channels will be split in half, which one less convolutional operation can be performed comparing to the previous module. The output will be processed separately on both channels, one is successively convolved using a 3×3 base convolutional ker-



(a) CSP layer

(b) CSPAM

Figure 4. (a) a single CSP layer; (b) our proposed CSPAM. None: no processing

-nel with a step size of two through three bottleneck structures, and its channels is divided into two halves each time, while the other without any operations. Data from one of the channels of the feature map is retained for stacking at the end after each pass through a bottleneck structure, which can be regarded as a dense residual structure.

After initial feature extraction in multiple CSPAMs, the output of the penultimate and third CSPAM from different shapes is saved and fed into the FPN structure for enhanced feature extraction. We decide to keep the original FPN structure with the fewest possible modifications and use a pooling module proposed in the next section to extract the deterministic features of the output of the last CSPAM.

B. Spatial Feature Pooling Module

It is crucial for model to enhance the ability to extract spatial information of features, therefore we propose a Spatial Feature Pooling Module called SFPM, which is shown in Figure 5, to replace the SPPBottleneck of CSPDarknet. The input image from the last CSPAM is set to $\boldsymbol{X} \in R^{C\times H\times W}$ and the $\boldsymbol{X}$ will first be fed into $\mathcal{F}_g$ and $\mathcal{F}_c$ separately for different feature extraction. $\mathcal{F}_g$ and $\mathcal{F}_c$ denote feature extractors. In $\mathcal{F}_c$, to preserve original features as much as possible, it just contains a 1×1 base convolutional layer, but reduce the number of channels of the input feature map to half the original number, which tentatively generates the statistics $\boldsymbol{X}_1 \in R^{C/2\times H\times W}$. In $\mathcal{F}_g$, it contains a 3×3 base convolutional layer, a 1×1 base convolutional layer, and a global pooling layer. To explore the importance of each channel, after initial extraction in the first two layers, we can get a feature tensor denoted $\boldsymbol{X}_2 \in R^{C\times H\times W}$. Subsequently, we separately utilize global max pooling operations on the $\boldsymbol{X}_2$. Specifically, for learning useful spatial information, in global pooling layer, three parallel pooling kernels of different sizes will be used to extract feature representations of different shapes, and the output $\boldsymbol{Z}_i \in R^{C/2\times H\times W},\ i \in [1,3]$ can be got:

$$\boldsymbol{Z}_i = \mathcal{F}\ (\boldsymbol{X}_2) \qquad (5)$$

Then we perform dimensional concat on the channel of the $\boldsymbol{Z}_i$ and aggregate them together with the following equation:

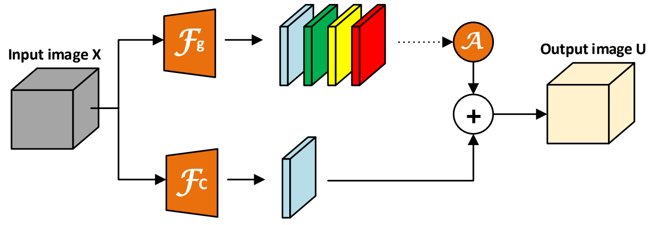


Figure 5. Overview of our proposed SFPM. $\mathcal{F}_g$ and $\mathcal{F}_c$ denote global pooling feature and convolutional feature extractors. $\mathcal{A}$: feature aggregation.

$$\boldsymbol{Q} = \mathcal{A}\ (\boldsymbol{Z}_i,\ \boldsymbol{X}_2) \qquad (6)$$

Finally, after performing dimensional concat on $\boldsymbol{X_1}$ with $\boldsymbol{Q}$, and a 1×1 base convolutional operation $\boldsymbol{\varphi_1}(\cdot)$, we obtain the most informative output $\boldsymbol{U} \in R^{C\times H\times W}$:

$$\boldsymbol{U} = \boldsymbol{\varphi_1}(\boldsymbol{X_1} + \boldsymbol{Q}) \qquad (7)$$

The output will be fed into the FPN structure and fused with other features from different shapes to further extract features and get three enhanced features for prediction at last.

C. Anchor-free

There are some known conflicts between classification and regression tasks when models using for object detection with original anchor-based detectors [11, 24, 14]. In terms of the overall computational complexity and latency, the anchor mechanism may become the potential bottleneck, especially moving large amount of predictions for each image between devices on some AI systems. Following the exploration of DenseBox [29] and YOLO, the performance of Anchor-free detectors [21, 39, 1] has been widely acknowledged in the past five years, which can achieve the same performance as anchor-based one while reducing the number of parameters.

Based on Anchor-free manner of YOLOX, we propose a new decoupled head, as shown in Figure 6, without the branch for IoU to lighten the model in terms of both speed and size. Firstly we can obtain three enhanced features of different shapes via the FPN structure and define the input feature maps as $\boldsymbol{I}$. Then the input $\boldsymbol{I}$ are separately processed through two parallel convolutional lines for the prediction of classification (Cls) and regression (Reg), which consists of two 3×3 base convolutional layers and two 1×1 convolutional layers. We take $\mathcal{F}_{tr}$ to be an operator to denote the above series of convolutional operations of each line, to get the final tensor $\boldsymbol{T}_i,\ i \in [1,2]$ for the branches to compute loss:

$$\boldsymbol{T}_i = \mathcal{F}_{tr}\ (\boldsymbol{I}) \qquad (8)$$

We still opt for BCE Loss for training classification branch, but we propose a new loss function Generalized Efficient IOU Loss called GEIOU Loss for regression branch because we infer the original use of IOU Loss may harm the performance. Compared to previous Losses [8, 45, 46, 30], GEIOU introduces Generalized Focal Loss [7] to achieves the great balance between hard and easy samples in the bounding box regression stage, and address the problem of learning distributed and qualified representation for dense object detection.

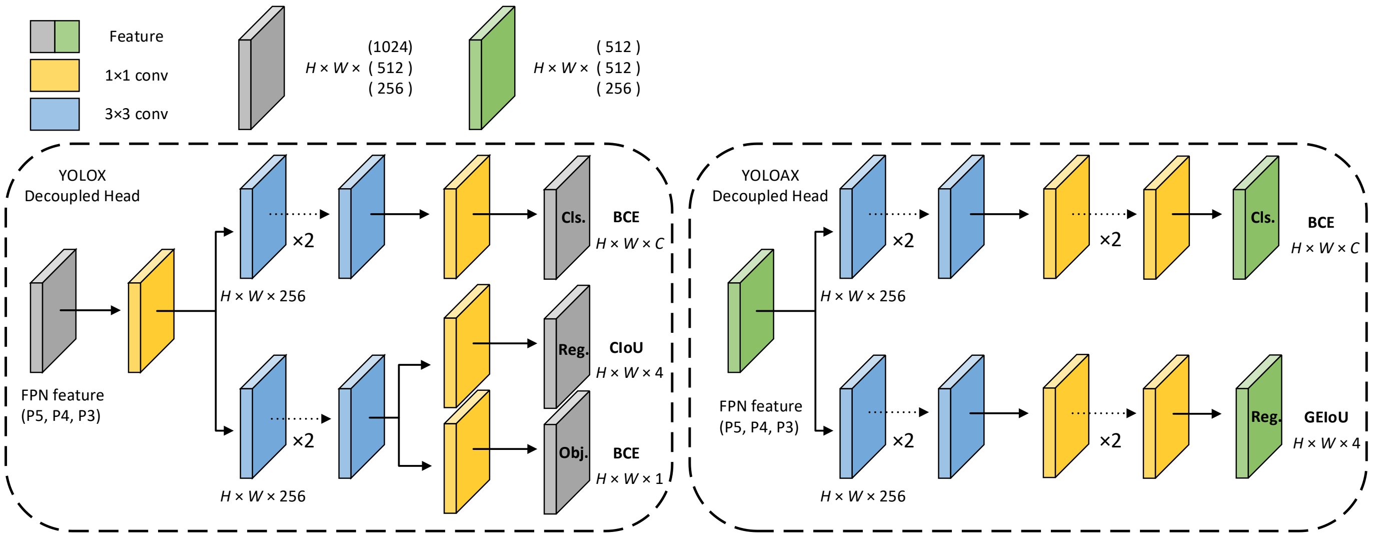


Figure 6. Illustration of the difference between YOLOX head and our proposed decoupled head. For each shape of FPN feature, we adopt two parallel branches with two 3×3 conv and two 1×1 conv layers each for taking classification and regression tasks without the Objectness branch.

In GEIOU, we divide the calculation of Reg loss into two parts: the first part uses EIOU Loss [8] to calculate the IOU between the prediction box B and the ground truth box B^{gt}; the latter part uses GFL to accurately locate box in arbitrary distribution and depict the flexible distribution in real data. The specific formulation of EIOU Loss can be defined as follows:

$$\begin{aligned}\mathcal{L}_{EIOU} &= \mathcal{L}_{IOU} + \mathcal{L}_{dis} + \mathcal{L}_{asp}\\ &= 1 - IOU + \frac{\rho^2(b,\ b^{gt})}{c^2} + \frac{\rho^2(w,\ w^{gt})}{(c^w)^2} + \frac{\rho^2(h,\ h^{gt})}{(c^h)^2},\end{aligned} \quad (9)$$

where b and b^{gt} represent the center points of each B and B^{gt}. w, h and w^{gt}, h^{gt} indicates the width and height of B and B^{gt}, respectively. c^w and c^h represent the width and height of the smallest outer box that covers both boxes.

Considering the problem that, if the number of high-quality anchors is much smaller than the number of poor samples with large regression errors, it may produce large gradient to affect the training phase when in a bounding box regression task. To tackle the problem, we reweight the EIOU Loss by using the value of GFL and get $\mathcal{L}_{GEIOU}$ as follows:

$$\mathcal{L}_{GEIOU} = GFL^{\beta} \cdot \mathcal{L}_{EIOU}, \quad (10)$$

where the specific formulation and derivation of GFL can be found in [7], and β can be seen as a controlling parameter to inhibit outliers. We also take ablation studies and observe significant improvements over counterparts trying other IOU Loss. More details are shown in Table 7.

D. Simple Task Assigner

Due to the fact that advanced label assignments have become an important progress for object detection, we opt for SimOTA [1] as our start point and optimize it a bit. And considering the excellence of the dynamic label assignments, we choose to combine it with TaskAlignedAssigner [25] in a weighted way, and propose a simple task assigner named STA for optimal transport problem.

For reducing the cost between ground truth box g_i and prediction p_j, we first add a new element, which is defined as E and denotes the samples that are excluded in the intersection part of g_i and p_j, to the cost function in SimOTA and use a coefficient μ to control the degree of scaling of the E. The new cost $\boldsymbol{C}_{ij}$ we proposed is calculated as:

$$\boldsymbol{C}_{ij} = \alpha\mathcal{L}_{cls} + \lambda\mathcal{L}_{reg} + \mu E, \quad (11)$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ are Cls loss and Reg loss between g_i and p_j. The α and λ are both balancing coefficient, and the μ is a controlling coefficient, an extremely large constant which we take the value of 100000, to force the detector to prefer samples within the intersection for matching in the process of minimizing cost.

Then inspired by the assigning strategy in TaskAlignedAssigner, which select positive samples based on the scores weighted by the Cls and Reg scores, we modify the $\boldsymbol{C}_{ij}$ as the follow equation:

$$\boldsymbol{C}_{ij} = \mathcal{L}_{cls}^{\alpha} \cdot \mathcal{L}_{reg}^{\lambda} + \mu E \quad (12)$$

For each g_i, the degree of alignment can be measured by multiplying the two weighted loss, and finally we select the top k prediction p_j with the least cost as its positive samples. As shown in Table 5, the STA show its power, which raises the performance of our detector from 50.7% AP to 51.3% AP. We also conduct serval ablation studies to test the power of STA, and more details please refer to the next section.

Ⅳ. EXPERIMENTS

We will conduct experiments and verify the effectiveness of our proposed object detection methods on COCO dataset [49] and PASCAL VOC 2012 test set [50]. All models of object detection in our experiments were not pre-trained, which were trained from scratch. We first used 2017 train set for training our detector and then used 2017 val set to validate and choose the optimal hyperparameter combinations. We show the high performance of our object detector YOLOAX on the 2017 test set at last. Experimental settings and detailed training parameter setup are described in the following section.

A. *Experimental Settings*

We opt for YOLOX with CSPDarknet as our baseline and design a basic model called YOLOAX for normal GPU. All the models and our proposed methods are trained on two parallel V100 GPU. Unless otherwise specified, our models are all trained and tested at 640 × 640 resolution on the MS-COCO dataset. Only the category labels of the images are used as annotations without any prior information.

To optimize our model, we experientially choose the SGD with a momentum of 0.937 as optimizer and set the total training process to 300 epochs. After several experimental comparisons, the learning rate is initially set to 1e-2 with warmup during the early processes of training, while reducing to 1/100 of last in a way of cosine annealing for every 30 epochs. And we employ a weight decay of 5e-4 and set the mini-batch size to 16. As for strong data augmentation strategies, we adopt the Mix-up and Mosaic implementations in our training but close it for the last 30 epochs to boost the model's performance. Nevertheless, during the training process, we found the benefit that suitable data augmentation strategies bring to model varies across different size of models. Therefore, when training small models such as YOLOAX-S, we weaken the Mosaic augmentation and remove the Mix-up. Such a modification, as demonstrated in Table 4, boost the performance of YOLOAX-S from 41.9% AP to 42.1% AP.

B. *Ablation Study*

Proposed modules To compare fairly, we adopt the exact backbone of YOLOX with CSPDarknet and the FPN structure. For exploiting the power of attention mechanism, we propose a cross-stage partial attention module called CSPAM based on channel attention, which can significantly highlight the most interested regions and help models improve the ability to learn the most important image representations. We set our CSPAM into the YOLOX-S to test the performance of it and then take ablation study on COCO and PASCAL VOC 2012 dataset to compare. The results of studies in Table 1,2 clearly verify the effectiveness of our CSPAM, which raises the detector from 40.5% AP to 40.8% AP on MS COCO and from 81.5% AP to 82.6% AP on PASCAL VOC with fewer parameters.

Model	CSPAM	AP (%)	Parameters	GFLOPs
YOLOX-S		40.5	9.0M	26.8
	✓	40.8	7.8M	24.2
improvement	-	**+0.3**	**-1.2**	**-2.6**

Table 1. Ablation study on our CSPAM on COCO.

Model	CSPAM	AP (%)	Parameters	GFLOPs
YOLOX-S		81.5	9.0M	26.8
	✓	82.6	7.8M	24.1
improvement	-	**+1.1**	**-1.2**	**-2.6**

Table 2. Ablation study on our CSPAM on PASCAL VOC.

Backbones We also follow YOLOX's scaling rule to design models of YOLOAX series (S, M, L, X). All the models are tested with image scale of 640 pixels and batch = 1 on two parallel V100 GPUs. As demonstrated in Table 3, our models show the consistent performance improvement over YOLOX series by about ~5.0% to ~3.0% AP on MS COCO without excessive inference cost.

Model	AP (%)	#Param.	GFLOPs	Latency
YOLOX-S	39.6	9.0M	26.8	9.8ms
YOLOAX-S	**42.1** (+2.5)	7.9M	26.3	10.9ms
YOLOX-M	46.4	25.3M	73.8	12.3ms
YOLOAX-M	**51.3** (+4.9)	22.2M	72.4	13.5ms
YOLOX-L	50.0	54.2M	155.6	14.5ms
YOLOAX-L	**53.8** (+3.8)	47.6M	152.7	15.3ms
YOLOX-X	51.2	99.1M	281.9	17.3ms
YOLOAX-X	**54.2** (+3.0)	87.0M	276.6	18.7ms

Table 3. Comparison of YOLOAX and YOLOX in terms of AP on COCO.

Model	Mosaic	Mix-up	AP (%)
YOLOAX-S	✓	-	42.1
	✓	✓	**41.9 (-0.2)**
YOLOAX-X	✓	-	52.8
	✓	✓	**54.2 (+1.4)**

Table 4. Effect of data augmentation across different size of models.

Proposed optimization methods In recent years, advanced label assignment is one of the most important progress of real-time object detection. For further optimize our model, based on the previous study SimOTA [1] and TaskAlignedAssigner [25], we thus design a simple task assigner named STA. It is obviously shown in Table 5 that STA raises the performance of our detector from 50.7% AP to 51.3% AP. We also conduct ablation studies on other detectors to test the performance when applying STA, where STA achieves performance improvements against all the corresponding counterparts.

Method	AP^{test}	AP^{test}_{50}	AP^{test}_{75}
base (FCOS)	41.5%	60.7%	45.0%
+STA	43.2% (+1.7)	63.8% (+3.1)	47.2% (+2.2)
base (RetinaNet)	39.1%	59.1%	42.3%
+STA	41.8% (+2.7)	60.4% (+1.3)	43.5% (+1.5)
base (YOLOX)	47.2%	65.4%	50.6%
+STA	47.5% (+0.3)	65.8% (+0.4)	50.8% (+0.2)
base (YOLOAX)	50.7%	68.9%	52.8%
+SimOTA	50.9% (+0.2)	69.2% (+0.3)	53.2% (+0.4)
+STA	**51.3%** (+0.6)	**69.5%** (+0.6)	**53.3%** (+0.5)

Table 5. Ablation studies on proposed STA on COCO.

Model	Backbone	#Param.	FLOPs	Size	FPS	AP^{test} / AP^{val}	AP^{test}_{50}	AP^{test}_{75}
PPYOLOE-S [57]	CSPResNet	7.9M	17.4G	640	208	43.1% / 42.7%	60.5%	46.6%
PPYOLOE-M [57]	CSPResNet	23.4M	49.9G	640	123	48.9% / 48.6%	66.5%	53.0%
PPYOLOE-L [57]	CSPResNet	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%
PPYOLOE-X [57]	CSPResNet	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%
YOLOv7 [2]	-	36.9M	104.7G	640	161	51.4% / 51.2%	69.7%	55.9%
YOLOv7-X [2]	-	71.3M	189.9G	640	114	53.1% / 52.9%	71.2%	57.8%
YOLOv7-W6 [2]	-	70.4M	360.0G	1280	84	54.9% / 54.6%	72.6%	60.1%
YOLOv7-E6 [2]	-	97.2M	512.2G	1280	56	56.0% / 55.9%	73.5%	61.2%
YOLOv7-D6 [2]	-	154.7M	806.8G	1280	44	56.6% / 56.3%	74.0%	61.8%
YOLOv7-E6E [2]	-	151.7M	843.2G	1280	36	**56.8% / 56.8%**	**74.4%**	**62.1%**
YOLOv5-S [4]	Modified CSP v5	7.2M	16.5G	640	156	- / 37.4%	-	-
YOLOv5-M [4]	Modified CSP v5	21.2M	49.0G	640	122	- / 45.4%	63.1%	-
YOLOv5-L [4]	Modified CSP v5	46.5M	109.1G	640	99	- / 49.0%	66.9%	-
YOLOv5-X [4]	Modified CSP v5	86.7M	205.7G	640	83	- / 50.7%	68.8%	-
YOLOR-CSP [12]	Modified CSP v5	52.9M	120.4G	640	106	51.1% / 50.8%	69.6%	55.7%
YOLOR-CSP-X [12]	Modified CSP v5	96.9M	226.8G	640	87	53.0% / 52.7%	71.4%	57.9%
YOLOX-S [1]	Darknet-53	9.0M	26.8G	640	102	40.5% / 40.5%	-	-
YOLOX-M [1]	Modified CSP v5	25.3M	73.8G	640	81	47.2% / 46.9%	65.4%	50.6%
YOLOX-L [1]	Modified CSP v5	54.2M	155.6G	640	69	50.1% / 49.7%	68.5%	54.5%
YOLOX-X [1]	Modified CSP v5	99.1M	281.9G	640	58	51.5% / 51.1%	69.6%	55.7%
YOLOAX-S	Modified CSPDarknet	7.9M	26.3G	640	119	42.1% / 42.0%	62.8%	47.7%
YOLOAX-M	Modified CSPDarknet	22.2M	72.4G	640	95	51.3% / 51.0%	69.5%	53.3%
YOLOAX-L	Modified CSPDarknet	47.6M	152.7G	640	84	53.8% / 53.5%	71.2%	57.2%
YOLOAX-X	Modified CSPDarknet	87.0M	276.6G	640	72	**54.2% / 54.2%**	**72.3%**	**58.4%**

Table 6. Comparison of state-of-the-art real-time object detectors.

Proposed loss for yolo_head Recently a trend for one-stage real-time detectors is to introduce two individual branch to predict the localization and estimate the quality of regression, which the quality of prediction facilitates the classification ability to improve detector performance. Thus, a more robust loss function is another important factor for real-time object detection process. Different from the manner of YOLOX, we only keep the branch for classification and regression with removing the Obj. branch. Normally, we still use BCE Loss for classification branch when training our detector. In terms of regression branch, we propose a Generalized Efficient IOU Loss called GEIOU, which can effectively reduce the regression loss of our object detector. To access the quality of our proposed loss, we select the YOLOX-M as our baseline and take some ablation studies on MS COCO dataset by trying different IOU Loss for the prediction of regression. In Table 7, it's clear that the significant improvements over other counterparts can be observed based on our proposed method, higher than EIOU and baseline by 0.4% AP and by 2.8% AP, respectively.

Method	AP^{test}	AP^{test}_{50}	AP^{test}_{75}
Baseline	47.2%	65.4%	50.6%
GIOU / DIOU	43.1% / 43.2%	62.3% / 62.5%	46.8% / 47.0%
CIOU / EIOU	45.1% / 46.3%	63.6% / 64.9%	48.1% / 48.8%
GEIOU(ours)	**47.4% (+0.2)**	**65.5% (+0.1)**	**50.7% (+0.1)**

Table 7. Comparison of our proposed GEIOU Loss and the counterparts in terms of AP (%) on COCO.

Method	AP^{test}	AP^{test}_{50}	AP^{test}_{75}
Baseline	64.4%	85.9%	72.0 %
GIOU / DIOU	64.9% / 66.1%	86.0% / 86.4%	72.3% / 73.1%
CIOU / EIOU	66.5% / 66.8%	86.8% / 87.0%	73.5% / 73.7%
GEIOU(ours)	**67.2% (+2.8)**	**87.3% (+1.4)**	**74.0% (+2.0)**

Table 7. Comparison of our proposed GEIOU Loss and the counterparts in terms of AP (%) on PASCAL VOC.

C. *Comparison with other state-of-the-arts*

We compare our proposed methods with state-of-the-art real-time object detectors for normal GPUs and the results are shown in Table 6. From the results in Table 6, we know that our proposed methods achieve the best trade-off comprehensively between speed and accuracy. If we compare YOLOAX series with YOLOX series, our method is on average about 15 fps faster and 3.1% more accurate on AP, especially our YOLOAX-X is 14 fps faster than YOLOX-X, improving about 24%. In addition, YOLOAX-M has 51.3% AP at frame rate of 95 fps, while PPYOLOE-L with the same AP has only 78 fps frame rate, reducing about 17%. In terms of parameter usage, YOLOAX-M is 57% less than PPYOLOE-L and 39% less than YOLOv7. If we compare YOLOAX-X with 72 fps computational speed to YOLOX-L with 69 fps computational speed, YOLOAX-X can improve AP by 4.1%. If YOLOAX-M is compared with YOLOv5-M of similar scale, the AP of YOLOAX-X on COCO is 4.6% higher. And then, as for the amount of computation and parameters, YOLOAX-X reduces 12% of parameters compared to YOLOX-X, but improves AP by 2.7%.

V. CONCLUSIONS

In this paper, we present some effective improvements to YOLOX, forming a high-performance object detector called YOLOAX. Equipped with the powerful modules we proposed, i.e., CSPAM and SFPM, YOLOAX attain a better ability to learn the most important image representations than YOLOX. And equipped with a new advanced label assigning strategy STA and optimal loss function GEIOU, YOLOAX achieves the best accuracy-speed trade-off over other state-of-the-art real-time counterparts. It is impressive that our YOLOAX-X get 54.2% AP on MS COCO at a real-time speed of 72.3 FPS, outperforming YOLOX-X by 2.7% AP.

ACKNOWLEDGMENT

This work is partially supported by Guangxi Science and T echnology Development Project (AB21075004; AB21220011; AB21220038).

References

[1]. Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021. 1, 5, 6, 7

[2]. Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 7464-7475. 1, 7

[3]. Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19. 2

[4]. glenn jocher et al. yolov5. https://github.com/ ultralytics/yolov5, 2021. 1, 2, 7

[5]. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020. 1, 2

[6]. Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141. 1

[7]. Li X, Wang W, Wu L, et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection[J]. Advances in Neural Information Processing Systems, 2020, 33: 21002-21012. 1, 4, 5

[8]. Zhang Y F, Ren W, Zhang Z, et al. Focal and efficient IOU loss for accurate bounding box regression[J]. Neurocomputing, 2022, 506: 146-157. 2, 4, 5

[9]. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587. 1, 2

[10]. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778. 2

[11]. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018. 1, 2

[12]. Wang C Y, Yeh I H, Liao H Y M. You only learn one representation: Unified network for multiple tasks[J]. arXiv preprint arXiv:2105.04206, 2021. 1, 2, 7

[13]. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448. 1, 2

[14]. Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28. 1, 2, 4

[15]. He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969. 1, 2

[16]. Park J, Woo S, Lee J Y, et al. Bam: Bottleneck attention module[J]. arXiv preprint arXiv:1807.06514, 2018. 2

[17]. Kim K, Lee H S. Probabilistic anchor assignment with iou prediction for object detection[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. Springer International Publishing, 2020: 355-371. 2

[18]. Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125. 1, 2

[19]. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30. 1

[20]. Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391. 2

[21]. Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 9627-9636. 2, 4

[22]. Du X, Lin T Y, Jin P, et al. Spinenet: Learning scale-permuted backbone for recognition and localization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11592-11601. 1

[23]. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022. 1

[24]. Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37. 2, 4

[25]. Feng C, Zhong Y, Gao Y, et al. Tood: Task-aligned one-stage object detection[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021: 3490-3499. 5, 6

[26]. Choromanski K, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[J]. arXiv preprint arXiv:2009.14794, 2020. 2

[27]. Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788. 1, 2

[28]. Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271. 1, 2

[29]. Huang L, Yang Y, Deng Y, et al. Densebox: Unifying landmark localization with end to end object detection[J]. arXiv preprint arXiv:1509.04874, 2015. 1, 4

[30]. Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas

Huang. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, MM '16, page 516–520, New York, NY, USA, 2016. Association for Computing Machinery. 4

[31]. Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9759-9768. 1

[32]. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020. 1, 2

[33]. Chen J, Lu Y, Yu Q, et al. Transunet: Transformers make strong encoders for medical image segmentation[J]. arXiv preprint arXiv:2102.04306, 2021. 1

[34]. Zhang X, Wan F, Liu C, et al. Freeanchor: Learning to match anchors for visual object detection[J]. Advances in neural information processing systems, 2019, 32. 1, 2

[35]. Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 734-750. 2

[36]. Chen Y, Yang T, Zhang X, et al. Detnas: Backbone search for object detection[J]. Advances in Neural Information Processing Systems, 2019, 32. 2

[37]. Zhu B, Wang J, Jiang Z, et al. Autoassign: Differentiable label assignment for dense object detection[J]. arXiv preprint arXiv:2007.03496, 2020. 1

[38]. Ma Y, Liu S, Li Z, et al. Iqdet: Instance-wise quality distribution sampling for object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 1717-1725. 1

[39]. Tian Z, Shen C, Chen H, et al. FCOS: A simple and strong anchor-free object detector[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(4): 1922-1933. 2, 4

[40]. Han K, Wang Y, Chen H, et al. A survey on vision transformer[J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(1): 87-110. 1, 2

[41]. Corbetta M, Shulman G L. Control of goal-directed and stimulus-driven attention in the brain[J]. Nature reviews neuroscience, 2002, 3(3): 201-215. 2

[42]. Guo M H, Xu T X, Liu J J, et al. Attention mechanisms in computer vision: A survey[J]. Computational visual media, 2022, 8(3): 331-368. 2

[43]. Choromanski K, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[J]. arXiv preprint arXiv:2009.14794, 2020. 2

[44]. Lu X, Li B, Yue Y, et al. Grid r-cnn[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7363-7372. 1

[45]. Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 658-666. 4

[46]. Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000. 4

[47]. Chen Y, Dai X, Chen D, et al. Mobile-former: Bridging mobilenet and transformer[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5270-5279. 1, 2

[48]. Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542. 2

[49]. Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014: 740-755. 5

[50]. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015. 5

[51]. Du X, Lin T Y, Jin P, et al. Spinenet: Learning scale-permuted backbone for recognition and localization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11592-11601. 2

[52]. Li Z, Peng C, Yu G, et al. Detnet: A backbone network for object detection[J]. arXiv preprint arXiv:1804.06215, 2018. 2

[53]. Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6569-6578. 2

[54]. Guo J, Han K, Wu H, et al. Cmt: Convolutional neural networks meet vision transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12175-12185. 2

[55]. Zhang H, Duan J, Xue M, et al. Bootstrapping ViTs: Towards liberating vision transformers from pre-training[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 8944-8953. 1, 2

[56]. Petit O, Thome N, Rambour C, et al. U-net transformer: Self and cross attention for medical image segmentation[C]//Machine Learning in Medical Imaging: 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 12. Springer International Publishing, 2021: 267-276. 2

[57]. Xu S, Wang X, Lv W, et al. PP-YOLOE: An evolved version of YOLO[J]. arXiv preprint arXiv:2203.16250, 2022. 7