Rethinking Features-Fused-Pyramid-Neck for Object Detection

Hulin Li

College of Traffic and Transportation, Chongqing Jiaotong University, Chongqing 400074, China
alan@mails.cqjtu.edu.cn

Abstract. Multi-head detectors typically employ a features-fused-pyramid-neck for multi-scale detection and are widely adopted in the industry. However, this approach faces feature misalignment when representations from different hierarchical levels of the feature pyramid are forcibly fused point-to-point. To address this issue, we designed an independent hierarchy pyramid (IHP) architecture to evaluate the effectiveness of the features-unfused-pyramid-neck for multi-head detectors. Subsequently, we introduced soft nearest neighbor interpolation (SNI) with a weight-downscaling factor to mitigate the impact of feature fusion at different hierarchies while preserving key textures. Furthermore, we present a feature adaptive selection method for downsampling in extended spatial windows (ESD) to retain spatial features and enhance lightweight convolutional techniques (GSConvE). These advancements culminate in our secondary features alignment solution (SA) for real-time detection, achieving state-of-the-art results on Pascal VOC and MS COCO. Code will be released at https://github.com/AlanLi1997/rethinking-fpn.

Keywords: Object detection · Feature pyramid · Feature misalignment · Detection-net architecture

1 Introduction

Bounding box regression for object detection and pixel-level boundary regression for instance segmentation are pivotal and challenging visual tasks. Unlike basic image classification [4] , detection or segmentation tasks often involve scenes with multiple objects of varying scales. Early deep learning-based object detection models began exploring real-time multi-scale object detection. RCNN [9] employed selective search to extract a large number of region proposals ($\sim$ 2k). Although the vast number of image patches (region proposals) might aid multi-scale detection, in practice, most scenes do not contain such a high number of objects to recognize. This approach was later deemed inefficient [8], as the proposed regions were derived from the image rather than features, necessitating a convolutional neural network (CNN) to extract features and predict classes for each proposal. A simplified RCNN architecture, as depicted in Fig. 1 (a), converts images of different scales into feature maps for object detection, largely

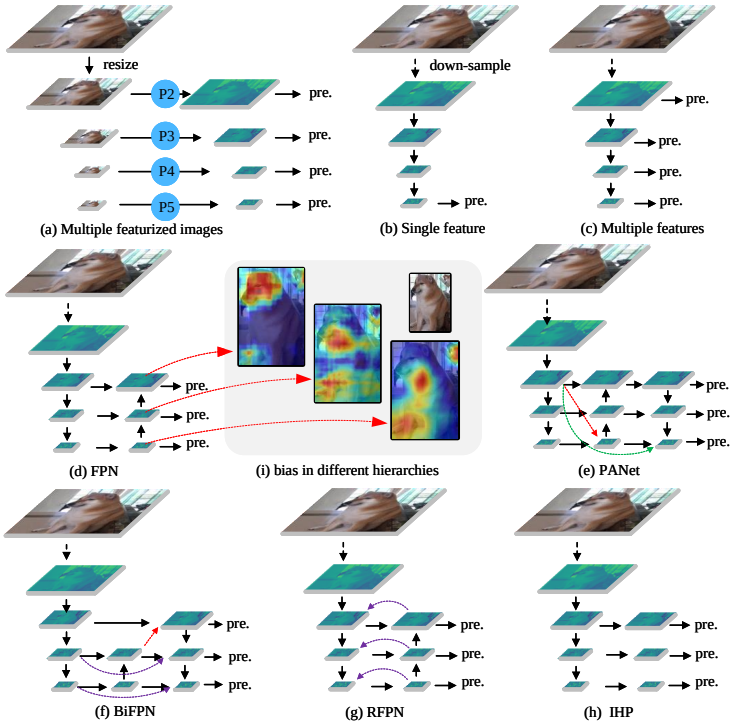

Fig. 1: The IHP and seven other typical neck architectures. The three heat maps are come from the third (P3), fourth (P4), and fifth (P5) hierarchies of the FPN-neck model, respectively. The area of interest for each hierarchies' features is highlighted in red; the darker the color, the higher the feature weight. It is evident that lower-level hierarchies prioritize local features, while higher-level hierarchies focus more on global features. In other words, there is a representation bias at different hierarchical levels. If these features are combined by point-to-point fusion without first addressing the misalignment issue, it may introduce noise rather than enrich semantics.

neglecting the real-time detection capability of models. YOLO [29] introduced the first real-time deep learning-based object detection model, which did not specifically focus on multi-scale detection, with a simplified structure shown in Fig. 1 (b). SSD [25] effectively combined the strengths of RCNN and YOLO, emphasizing the concept of multi-scale detection. It was the first model to achieve real-time multi-scale object detection using multi-scale features, with an architecture similar to Fig. 1 (c). SSD efficiently utilized feature maps of different scales from the VGG-16 backbone [32] to predict objects of varying scales. The exploration by early deep learning-based object detection models provided an important insight to the field: the efficient utilization of multi-scale features is crucial for enhancing the overall performance (accuracy-speed trade-off) of detectors. Consequently, FPN [21] introduced the feature pyramid-based multi-scale

feature fusion concept, which gradually became a structural paradigm for detectors. This led to numerous representative works enhancing multi-scale feature fusion, such as PANet [24], ASFF [23], BiFPN [34], and RFPN [28]. The performance of real-time detectors has improved with the implementation of feature fusion strategies. However, further advancements are hindered by the increasing complexity of these techniques. Before the introduction of FPN, object detectors primarily relied on structures without fusion [9, 25, 29], as depicted in Fig. 1 (a)-(c). With the advent of FPN, these detectors have shifted towards FPN-based paradigms or more intricate improvement methods [8, 20, 29–31], as illustrated in Fig. 1 (e)-(g). It is important to note that the FPN-like strategy should not be considered an obligatory component for all detectors. Representation learning suggests that models ultimately require refined representations rather than an abundance of complex representations. Under the FPN-like paradigm, feature maps of different hierarchies and different resolutions are combined at the element level (point-to-point), necessitating the expansion of low-resolution, high-level feature maps to match the resolution of low-level feature maps. However, there are inherent representation biases in feature maps at different hierarchical levels: low-level feature maps tend to retain representations of small objects (local features in small receptive fields), while high-level feature maps favor representations of large objects (global features in large receptive fields), as shown in Fig. 1 (i). The direct element-level fusion of these feature maps with representation bias may lead to partial destruction of representations and exacerbate the feature misalignment problem during fusion.

In visual models, the resolution of feature maps decreases progressively from low to high hierarchical levels of the feature pyramid, achieved through step-by-step downsampling. Typically, downsampling reduces the resolution (width $\times$ height) of feature maps at a square root rate, inherently filtering out some spatial information regardless of whether the number of features (channels) remains constant or increases. It is also noteworthy that the transition of feature information from spatial to channel dimensions primarily occurs during downsampling, underscoring the importance of this stage. Traditional downsampling methods operate in a single mode, such as convolution with a stride of 2 (learnable) or average pooling and max pooling with a stride of 2 (non-learnable). These single-mode approaches have persisted since the introduction of ResNet [12] without significant evolution, which merits further investigation. Additionally, the initial purpose of FPN was not only to demonstrate the advantages of feature fusion in addressing multi-scale detection challenges but also to appropriately reduce model complexity. In industries with limited hardware resources, complex models are often impractical [2] [10]. Therefore, we enhance the lightweight convolution technique of the GSConv [18] to achieve a preferable trade-off between accuracy and speed for real-time detection models.

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

1).We rethink the effectiveness of FPN-like paradigms for modern real-time detectors based on representation learning and identify issues with feature misalignment during element-level fusion.

2).We design an independent hierarchy pyramid architecture (IHP) without feature fusion in the neck to validate our findings. And the IHP achieves advanced results.

3).We introduce soft nearest neighbor interpolation (SNI) for up-sampling to mitigate feature misalignment during fusion.

4).We provide a feature adaptive selection method in extended spatial windows for downsampling (ESD) to enhance spatial feature retention during the downsampling stage. This method is plug-and-play and optimizes performance at a low cost.

5).We simplify and enhance the GSConv lightweight convolution (GSConvE) to offer more cost-effective options for models operating on edge devices with resource constraints.

6).We validate our approaches on Pascal VOC and COCO and present the SA solution. Existing lightweight real-time detection models can be optimized with the SA solution without bells and whistles.

2 Related Work

2.1 Multi-head detection and features fusion

The first-generation general detection models using deep learning consist of two main components: the backbone and the detecting-head [9, 29]. These models typically use final feature maps for prediction. SSD [25] introduced the use of multi-level feature maps for object detection, marking the beginning of multi-head detection and influencing subsequent real-time detector designs [31]. In the SSD model, each detecting-head directly uses raw feature maps from different levels of the backbone without any fusion. The key advantage of the multi-head detection method is the ability to predict objects of different scales using feature maps with varying receptive fields. High-level feature maps, with their wide global receptive fields, are beneficial for identifying large objects, while low-level feature maps, with refined local receptive fields, are helpful for identifying small objects. FPN introduced a fusion scheme for multi-level feature maps to enhance detection accuracy, setting the trend for feature fusion and becoming a common practice. This led to the development of Backbone-Neck-Head architectures. More complex variants based on FPN, such as BiFPN [34] and PANet [24], have since been proposed, enabling the use of more features for prediction. However, as prediction accuracy improved, networks became increasingly complex due to additional fusion layers. Some approaches have attempted to bypass feature fusion schemes by exploring new perspectives on multi-scale detection challenges, such as optimizing multi-scale training strategies [33] or using dilated convolutions to capture multi-scale receptive fields [19]. But these methods have gradually lost their competitiveness with the continuous optimization of classification-backbone feature extraction capabilities and the ongoing development of feature fusion techniques [1].

2.2 Lightweight

The choice between cloud computing or edge computing for a model depends on its number of parameters and floating-point operations (FLOPs). Therefore, reducing the number of parameters or FLOPs is a primary focus in lightweight model studies. Direct lightweight approaches include reducing the depth (number of layers) or width (number of neurons/filters) of the model, as seen in models like YOLO-fast/tiny/nano [1,14–16,31,35]. Additionally, sparse computing techniques, such as depth-wise separable convolution used in MobileNets [13] and ShuffleNet [39], can also be employed. However, low-depth networks often suffer from underfitting due to insufficient nonlinear representation capabilities. Therefore, reducing the number of network layers may not always be a cost-effective way to lighten the model.

3 Secondary Features Alignment Solution

Key contents of this work are described in detail in this section. Specifically, there are the IHP structure to directly demonstrate features unalignment problem by abandoning fusion, the SNI to alleviate unaligned during features fusion by point-to-point, the ESD to enhance spatial-features capture during the downsampling stage, and the GSConvE to improve the performance between the accuracy and speed of lightweight models.

3.1 Independent hierarchy pyramid architecture

The introduction of FPN brought all real-time detectors into the realm of feature fusion. However, we have identified a significant issue arising from the point-to-point fusion of different level features – partial local features become unaligned. This fusion process is akin to adding noise when unaligned features are combined, as features unrelated to the target space are forcibly integrated, resulting in spatial disarray. For instance, as depicted in Fig. 2, adding the head-feature of a puppy to the butt creates an undefined breed when fed to the detection-head. This scenario is particularly evident when the puppy occupies fewer pixels. Typically, nearest neighbor interpolation is used to rapidly increase the resolution of high-level feature maps to match that of low-level feature maps, followed by fusion through methods such as element-wise addition, channel concatenation, or weighted sum. However, downsampling tends to be nonlinear and irreversible, leading to inconsistencies in spatial features among the additional features generated by up-sampling. Although BiFPN [34] offers a learnable fusion method, it has not been widely adopted by recent state-of-the-art real-time detectors, such as YOLOv7 [35] or YOLOv8 [15], due to the increased complexity and lack of demonstrated comprehensive performance improvements in these models.

The IHP adopts a radical approach by abandoning all fusions in the neck, fundamentally circumventing the features misalignment problem and streamlining the structure. By leveraging the inherent benefits of the multi-head detection

mode, the IHP directly predicts objects of different sizes using feature maps of different levels. It is important to note that there are significant differences between IHP and the SSD architecture: SSD directly uses different level features from the backbone to detect objects but the IHP introduces a bottleneck convolution module before prediction to filter the features. The advantage of this approach is that it allows the model to independently learn features for a specific scale branch without affecting features at other levels. Directly stacking convolutional blocks at different levels of the backbone can lead to interference between different levels, which we believe might be the primary reason why using a very deep backbone in object detection models can result in performance degradation. For instance, DSSD [6] showed a performance drop when ResNet-101 was used as the backbone instead of VGG-16 in the SSD detector. In Part I of Tab. 1, all baselines of coupled-head YOLOs demonstrated improved results when using the IHP. Features fusion often overlooks the potential introduction of noise when augmenting semantic richness for prediction. While the IHP may appear to reduce semantic information, it effectively harnesses the advantages of multi-head detection to circumvent the misalignment problem. This strategic utilization of multi-head detection is the key factor behind the competitive results achieved by the IHP. Unfortunately, unlike localization-classification-coupled-head detectors [14,29–31,35], the IHP did not achieve the same positive reaction to localization-classification-decoupled-head detectors [15, 16]. Decoupled-head detectors address a key challenge faced by coupled-head detectors: the disparity in the interests of classification and localization tasks, both of which are often performed using the same layer for prediction [36]. This approach is akin to forcibly fusing misaligned features for prediction. Decoupled-head detectors overcome this limitation by independently predicting classification and localization tasks through separate branches. This enables them to learn beneficial representations from redundant features, which is not possible in coupled-head detectors. Consequently, while FPN strategies are still applicable to some extent in decoupled-head detectors, the features misalignment problem still needs to be addressed.

3.2 Soft nearest neighbor interpolation

In detection or segmentation models, different-level feature fusion typically occurs after the up-sampling of feature maps, with nearest neighbor interpolation and transposed convolution being popular up-sampling methods. However, the resolution expansion resulting from up-sampling in detection or segmentation differs significantly from tasks such as generation or super-resolution. In detection, the expanded feature maps represent abstract high-level semantic features rather than raw image detail information. Transposed convolution introduces increased computation and latency. To address the features misalignment problem with minimal cost while maintaining the speed of nearest neighbor interpolation, we explore a solution. Nearest neighbor interpolation is akin to average unpooling but is considered a "hard" operation. By softening this operation, similar to the transformation of SoftMax to the Max function, we aim to mitigate the

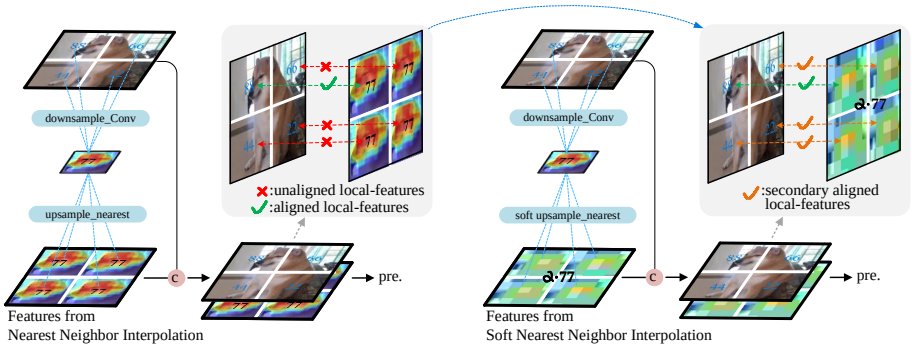

Fig. 2: Illustrations of the features misalignment in fusion and the SNI. These numbers, 22-88, are just markers of different local-features not real feature values.

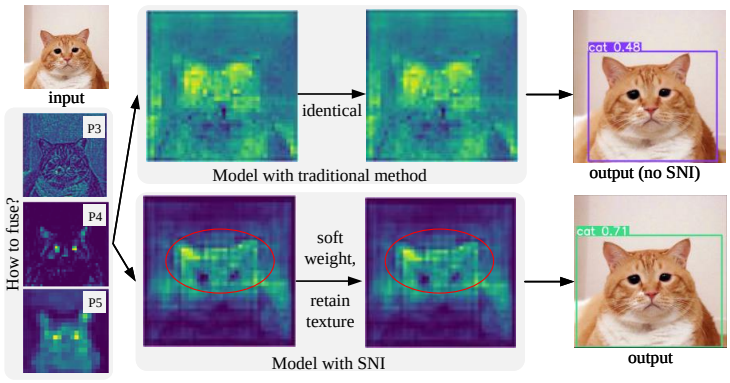

Fig. 3: A comparison of the SNI and traditional method. This result could be reproduced by using SNI source code (model:Yolov5n-panet). The same size scaling before and after up-sampling is for intuitive comparison.

"hard" features misalignment issue. To achieve this, we introduce a soft factor, denoted as α, for the nearest neighbor interpolation during up-sampling:

$$\boldsymbol{Y} = \alpha \cdot \boldsymbol{f}(\boldsymbol{X}), \alpha = \frac{Resolution\boldsymbol{X}}{Resolution\boldsymbol{Y}} \tag{1}$$

In Eq. (1), the resolution of the high-level feature maps ($\boldsymbol{X}$) and low-level feature maps ($\boldsymbol{Y}$) are denoted as $Resolution\boldsymbol{X}$ and $Resolution\boldsymbol{Y}$, respectively. The nearest neighbor interpolation operation, represented by $\boldsymbol{f}$, is illustrated in Fig. 2. The SNI adjusts the influence of high-level semantic features on low-level features based on the zoom factor for feature maps. Specifically, as the zoom factor increases, the impact of high-level semantic features on low-level features weakens. Unlike SoftMax, the SNI does not convert outputs into probabilities but rather mitigates misalignment without additional cost when integrating high-

level features with low-level features through point-to-point fusion. SNI softens feature weights while preserving key textures, thereby reducing the mutual influence of features from different hierarchical levels during fusion, as qualitatively demonstrated in Fig. 3. This approach achieves the secondary (auxiliary) features alignment (SA) and lowers the complexity for the model to learn from the fused representations. In contrast, traditional methods that directly merge features from different hierarchical levels can cause unstable competitive effects due to representation biases, increasing the learning difficulty for the model. For example, if the three heatmaps in Fig. 1 (i) are directly fused, should the red areas cover the entire image? Clearly, the answer is no. If such a situation occurs, it indicates a failure in prior feature extraction, as the objective of a model is to capture and focus on useful features. This is also why models under the FPN-like paradigm typically require several additional convolutional blocks to further filter features after fusing and before predicting.

In Part II of Tab. 1, all tested state-of-the-art models demonstrate accuracy improvements solely by incorporating the SNI, confirming the presence of features misalignment in FPN-based models. Notably, for YOLOv6 [16], a detector equipped with various techniques tailored for industry applications, the AP is enhanced by up to 2 percentage points just with the adoption of SNI, while reducing computational cost and inference latency. We achieved significant optimization results in the original YOLOv6 by simply replacing transposed convolution with SNI, highlighting the efficacy of this approach.

3.3 Feature adaptive selection in extended spatial windows

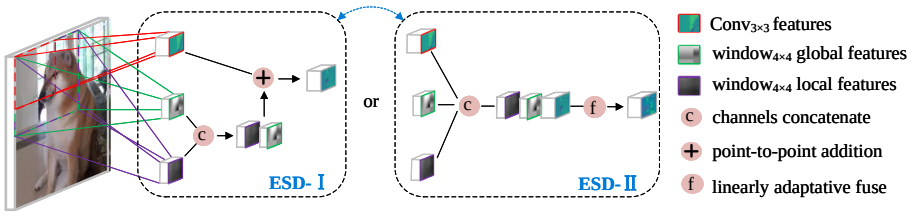

Fig. 4: Illustrations of the ESD-I and ESD-II. It must be mentioned that although the size of the extended window does not have to be same in the ESD-I, pooling is a non-learnable operation, so the features from an unequal window may become noise for other and bring the trouble of the features misalignment again. The ESD-II is free on size of the extended window because of the the learnable fusion like the SPPNet [11] for the YOLOv3.

Feature downsampling is a critical technique in vision models based on deep learning, involving the progressive reduction of feature map resolution. In vision representation learning, models typically rely on low-resolution, high-level

feature maps for prediction rather than directly processing high-resolution raw images. However, excessive downsampling can result in significant loss of spatial details [12]. Recent detectors typically limit the number of downsampling operations to no more than five times, leading to a spatial resolution reduction of 2^5 times, as demonstrated in YOLOs [1, 7, 14–16, 31, 35, 37] and Swin Transformer [26]. Common downsampling methods include convolution and pooling, with models typically employing only one of these methods. We propose the feature adaptive selection method in extended spatial windows for downsampling (ESD) to enhance spatial information retention capabilities. ESD consists of two nonlinear branches and one linear branch. In the nonlinear branches, a norm convolutional layer and an extended window max-pooling layer enhance local feature capturing, while extended window average-pooling enhances global feature capturing. Subsequently, linear features are merged with nonlinear features. Two fusion modes, ESD-I and ESD-II, are designed for lightweight and norm models, respectively. The difference lies in the feature fusion stage, as illustrated in Fig. 4. ESD-I employs element-wise addition for merging features, offering low computational cost suitable for lightweight models, while ESD-II utilizes learnable linearly adaptive fusion, slightly increasing computational cost but suitable for norm models. Importantly, local and global features of the extended window are sampled using simple pooling techniques, preserving input information to a significant extent. This reduces information loss during downsampling and enables subsequent layers to learn partial representations from the previous layer, akin to indirectly realizing a hidden shortcut connection reminiscent of ResNet [12].

3.4 Lightweight GSConv enhancement

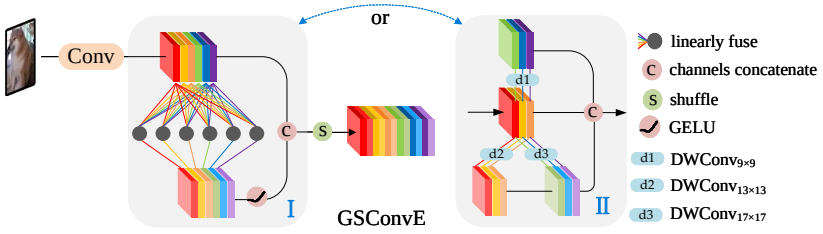

Fig. 5: Illustrations of the GSConvE-I and II.

GSConv [18] was introduced to address the channel interaction limitations of depthwise separable convolution by blending features from both vanilla convolution and depthwise convolution. The original GSConv consists of three fundamental components: a 3×3 vanilla convolution, a 5×5 depthwise convolution, and a feature shuffle operation. Building upon this, we propose two evolutions

of GSConv, namely GSConvE-I and GSConvE-II, tailored for norm models and lightweight models, respectively. GSConvE-I is designed for norm models, where intermediate feature maps on the auxiliary branch are derived from dense linear mappings, while output feature maps originate from sparse linear mappings. This approach maintains feature richness without significantly increasing computational cost. On the other hand, GSConvE-II is geared towards lightweight models. It incorporates three depthwise convolution auxiliary branches with kernel sizes of 9×9, 13×13, and 17×17, respectively. By utilizing larger kernel sizes, this variant can directly capture larger receptive fields and global features with fewer layer accumulations, offering high cost-effectiveness for lightweight models that prioritize reduced FLOPs and layer count. In GSConvE-I, we explore the removal of the batch normalization layer on the depthwise convolution branch while retaining the batch normalization layer on the main branch, similar to ConvNeXt [27]. This adjustment avoids the vanishing gradient problem and simplifies computation, leading to improved prediction accuracy.

4 Experiments

Real-time detectors are highly favored in industry due to their hardware-friendly nature and the optimal balance they strike between accuracy and speed. In our study, we selected popular YOLO models—YOLOv3 [31], YOLOv4 [1], YOLOv5 [14], YOLOv6 [16], YOLOv7 [35], and YOLOv8 [15]—as baselines to evaluate the efficacy of the SA solution in real-time detection scenarios (SYOLO).

4.1 Datasets

We conducted trials on the Pascal VOC dataset [5] and the MS COCO dataset [22]. The Pascal VOC 07+12 dataset was utilized for ablation experiments, wherein models underwent training on the train&val07+12 set and were subsequently evaluated on the test07 set. On the other hand, the MS COCO dataset was employed for comparative experiments, with models trained on the train2017 set and evaluated on the val2017 set. To ensure fairness and consistency, all released models were trained using the default set of hyperparameters and without utilizing pre-trained weights. More details of hyperparameters are in the support material.

4.2 Ablation studies

In Tab. 1, we present the results of ablation experiments conducted on six baselines [1, 14–16, 31, 35], focusing on the impact of the IHP, SNI, ESD. The lightweight experiment results of GSConvE are reported in Tab. 2. All baselines retained their original neck architecture, as we believe the competitive neck has been carefully chosen in the original works. Performance metrics highlighted in green indicate performance gains, while those in red indicate losses. It is crucial

to address an often-overlooked aspect during the verification process: the inconsistency between reported evaluation accuracy and actual usage. Frequently, detectors are evaluated with a very low confidence threshold, such as 0.001, which results in a large number of prediction boxes and inflated evaluation accuracy. However, in real-world detection scenarios, the confidence threshold is typically set much higher, around 0.25 or above. This filtering process ensures that higher-quality bounding boxes are utilized for practical purposes. The discrepancy between evaluation settings and real-world usage can create a misleading impression of accuracy. Therefore, it is essential to maintain consistency in configuration between evaluation and application. While this may result in a significant drop in evaluation accuracy, as shown in Tab. 3, it provides a more realistic assessment of model performance.

Table 1: Ablation experiments of the IHP, SNI and ESD-I/II on VOC 07+12.

Models	**Size**	**Param.**(M)	**FLOPs**(G)	$\mathbf{AP}_{50}$(%)	**AP**(%)	$\mathbf{Latency}^{T4}_{b=16}$(ms)
baselines						
Yolov3-t [31]	640	8.71	13.0	58.0	27.3	3.8
Yolov4-t [1]	640	5.91	16.1	62.6	32.2	4.4
Yolov5-n [14]	640	1.79	4.2	74.0	47.0	3.4
Yolov6-n [16]	640	4.30	11.1	80.2	58.8	3.5
Yolov7-t [35]	640	6.06	13.3	78.4	54.6	5.0
Yolov8-n [15]	640	3.01	8.1	79.8	59.7	3.7
Part I: IHP (ours)						
Yolov3-t-ihp	640	9.33	15.0	59.9(↑1.9)	29.1(↑1.8)	3.8
Yolov4-t-ihp	640	6.63	16.0	63.8(↑1.2)	33.9(↑1.7)	4.1
Yolov5-n-ihp	640	1.66	3.9	74.8(↑0.8)	48.6(↑1.6)	**3.2**
Yolov6-n-ihp	640	4.37	10.6	79.4(↓0.8)	57.4(↓1.4)	3.7
Yolov7-t-ihp	640	6.77	14.5	80.7(↑2.3)	55.3(↑0.7)	4.7
Yolov8-n-ihp	640	2.99	8.1	79.3(↓0.5)	59.1(↓0.6)	3.7
Part II: SNI (ours)						
Yolov3-t-sni	640	8.71	13.0	58.6(↑0.6)	28.0(↑0.7)	3.8
Yolov4-t-sni	640	5.91	16.1	63.4(↑0.8)	33.0(↑0.8)	4.4
Yolov5-n-sni	640	1.79	4.2	74.5(↑0.5)	47.7(↑0.7)	3.4
Yolov6-n-sni	640	4.28	11.0	**81.9**(↑1.7)	60.8(↑2.0)	3.4
Yolov7-t-sni	640	6.06	13.3	81.5(↑3.1)	57.9(↑3.3)	5.0
Yolov8-n-sni	640	3.01	8.1	80.3(↑0.5)	60.1(↑0.4)	3.7
Part III: ESD-I (ours)						
Yolov3-t-esd-I	640	11.8	22.4	69.2(↑11.2)	38.5(↑11.2)	4.6
Yolov4-t-esd-I	640	5.91	16.1	63.1(↑0.7)	32.9(↑0.7)	4.4
Yolov5-n-esd-I	640	1.78	4.2	75.4(↑1.4)	48.0(↑1.0)	3.8
Yolov6-n-esd-I	640	4.30	11.1	80.5(↑0.3)	58.8	3.4
Yolov7-t-esd-I	640	6.83	14.6	79.2(↑0.8)	55.3(↑0.7)	5.2
Yolov8-n-esd-I	640	3.22	8.1	80.1(↑0.3)	59.8(↑0.1)	3.8
ESD-II (ours)						
Yolov3-t-esd-II	640	8.78	14.8	63.0(↑5.0)	32.1(↑4.8)	4.9
Yolov4-t-esd-II	640	5.93	16.6	63.9(↑1.5)	33.5(↑1.3)	4.7
Yolov5-n-esd-II	640	2.32	4.7	75.7(↑1.7)	48.9(↑1.9)	3.6
Yolov6-n-esd-II	640	4.32	11.1	80.4(↑0.2)	58.8	3.5
Yolov7-t-esd-II	640	7.69	15.7	79.3(↑0.9)	55.2(↑0.6)	5.9
Yolov8-n-esd-II	640	3.18	8.5	80.4(↑0.6)	59.9(↑0.2)	4.0

Table 2: Ablation experiments of the GSConv and GSConvE-I/II on VOC 07+12.

Models	**Size**	**Param.**(M)	**FLOPs**(G)	**AP**$_{50}$(%)	**AP**(%)	**Latency**$^{T4}_{b=16}$(ms)
baselines(GSConv [18])						
Yolov3-tiny-gsconv	640	4.61	8.3	69.9	41.1	3.9
Yolov4-tiny-gsconv	640	3.42	12.4	73.4	45.2	4.6
Yolov5-n-gsconv	640	1.51	3.9	74.2	47.9	3.5
Yolov6-n-gsconv	640	4.27	11.0	80.4	59.1	3.4
Yolov7-tiny-gsconv	640	5.00	11.4	78.4	55.3	5.6
Yolov8-n-gsconv	640	2.74	7.8	80.9	61.2	3.7
GSConvE-I (ours)						
Yolov3-t-gse-I	640	7.86	11.6	71.7(↑1.8)	43.8(↑2.7)	3.9
Yolov4-t-gse-I	640	4.88	14.3	74.9(↑1.5)	47.8(↑2.6)	4.7
Yolov5-n-gse-I	640	1.76	4.2	75.8(↑1.6)	50.1(↑2.2)	3.5
Yolov6-n-gse-I	640	4.30	11.1	80.1(↓0.3)	58.8(↓0.3)	3.6
Yolov7-t-gse-I	640	6.21	13.5	79.1(↑0.7)	56.4(↑1.1)	5.8
Yolov8-n-gse-I	640	2.96	8.1	81.2(↑0.3)	61.3(↑0.1)	3.8
GSConvE-II (ours)						
Yolov3-t-gse-II	640	2.85	6.3	70.1(↑0.2)	41.4(↑0.3)	4.0
Yolov4-t-gse-II	640	2.36	10.8	75.6(↑2.2)	48.0(↑2.8)	4.6
Yolov5-n-gse-II	640	1.48	3.9	76.3(↑2.1)	51.4(↑3.5)	3.5
Yolov6-n-gse-II	640	4.28	11.1	80.3(↓0.1)	58.8(↓0.3)	3.5
Yolov7-t-gse-II	640	4.33	10.7	79.1(↑0.7)	56.2(↑0.9)	5.8
Yolov8-n-gse-II	640	2.67	7.7	79.9(↓1.0)	60.4(↓0.8)	3.8

Table 3: A contrast of the different confidence threshold for the SYOLO on COCO.

Confidence threshold	**AP**(%)	**AP**$_s$(%)	**AP**$_m$(%)	**AP**$_l$(%)
0.001(valuation)	53.1	35.5	58.7	69.9
0.25(application)	48.7(↓4.4)	29.7(↓**5.8**)	54.5(↓4.2)	66.5(↓3.4)

4.3 Comparasion experiments

In Tab. 4 and Tab. 5, we compare state-of-the-art real-time detectors and SYOLO (architecture in Tab. 7) on the Pascal VOC and COCO benchmarks. In Tab. 6, we provide additional comparisons of state-of-the-art detectors on COCO [22], including both real-time (CNNs) and non-real-time (Transformers/CNNs) models. Parameters and FLOPs metrics significantly higher than those of real-time detector models are highlighted in red.

In Tab. 4, SNI increased the AP of YOLOv7 by 0.9 percentage points without any additional computational or memory costs. SYOLO, utilizing the SA solution, achieved the highest accuracy among competitive models on the Pascal VOC benchmark. In Tab. 5, SYOLO also achieved the highest accuracy among competitive models on the MS COCO benchmark at input resolutions of 416×416 and 640×640. Positive performance improvements across different benchmarks demonstrate the effectiveness of the SA solution for real-time object detection.

Table 4: Comparasions of state-of-the-art real-time detectors on VOC 07+12.

Models	**Year**	**Neck**	**Size**	**Param.**(M)	**FLOPs**(G)	**AP**(%)	**Latency**$^{T4}_{b=16}$(ms)
Yolov3-spp [31]	2018	FPN	640	61.60	154.9	60.5	39.4
Yolov4 [1]	2020	PANet†	640	52.57	119.3	66.2	31.0
Yolov5 [14]	2022	PANet†	640	46.21	108.0	63.9	30.1
YoloX [7]	2021	PANet†	640	53.79	153.7	66.7	40.2
Yolov6 [16]	2022	PANet†	640	65.81	159.1	69.2	29.5
Yolov7 [35]	2022	PANet†	640	36.58	103.5	68.2	23.2
Yolov8 [15]	2023	PANet†	640	43.62	164.9	69.3	31.6
SYolov7(ours)	2023	SA-PANet††	640	36.58	103.5	69.1	23.2
SYolo(ours)	2023	SA	640	57.23	142.4	**69.6**	29.3

† Using simplified PANet: the red and green shortcuts are cancelled in the Fig. 1 (e).
†† Using SNI replace the nearest neighbor interpolation to up-sampling in the PANet†.

Table 5: Comparisions of state-of-the-art real-time detectors on COCO.

Models	**Size**	**Param.**(M)	**FLOPs**(G)	**AP**(%)	**AP**$_s$(%)	**AP**$_m$(%)	**AP**$_l$(%)	**Latency**$^{T4}_{b=16}$(ms)
YoloX-t [7]	416	5.1	6.5	32.8	-	-	-	4.3
SYolo-n(ours)	416	4.8	12.0	**35.9**	**14.3**	**39.2**	**54.4**	4.3
YoloX-t [7]	640	5.1	15.4	34.7	-	-	-	7.2
Yolov5-s [14]	640	7.2	16.5	37.4	-	-	-	9.8
Yolov6-n [16]	640	4.7	11.4	37.5	-	-	-	4.3
Yolov7-t [35]	640	6.2	13.8	38.7	18.8	42.4	51.9	5.3
Yolov8-n [15]	640	3.2	8.7	37.3	-	-	-	4.1
SYolo-s(ours)	640	6.6	15.7	**40.8**	**21.1**	**45.3**	**56.9**	5.8

5 Conclusion

We explore the nature of representation learning and identify that the fusion of different level feature maps faces the issue of feature misalignment. To address this, we introduce the SNI. Additionally, we propose the ESD to mitigate spatial information loss during downsampling and further optimize lightweight convolutional methods for computational efficiency in lightweight models. Based on these advancements, we propose the SA solution for real-time detection, where all tested detectors surpass their original performance and achieve state-of-the-art results.

Object detection has significantly benefited from the FPN-like paradigm in past few years. However, recent developments indicate that newer technologies, including innovative architectures and training strategies, may offer superior performance for modern detectors, making the FPN-like paradigm potentially limiting. In many cases, the effectiveness of the FPN-like paradigm can be enhanced through the application of SNI to address feature misalignment issues. Fully addressing the challenge of misalignment during feature fusion in existing vision models requires further research efforts.

Table 6: Comparisions of state-of-the-art detectors on COCO.

Models	**Size**	**Param.**(M)	**FLOPs**(G)	**AP**(%)	**AP**$_s$(%)	**AP**$_m$(%)	**AP**$_l$(%)	**Latency**$_{b=1}^{V100}$(ms)
Transformers/CNNs (non-real-time, pre-trained)								
FPN [21]	-	56.4	145.8	36.8	17.5	38.7	47.8	-
BiFPN [34]	1024	21.0	55.0	49.3	-	-	-	42.8
ASFF [23]	800	-	-	43.9	27.0	46.6	53.4	34.0
PANet [24]	-	-	-	51.0	32.6	53.9	64.8	-
RFPN [28]	-	-	-	51.3	31.7	54.6	64.8	-
DETR-Def. [40]	-	40.0	173.0	43.8	26.4	47.1	58.0	-
DETR-Dyn. [3]	-	-	-	47.2	28.6	49.3	59.1	-
DETR-DND [17]	1333	48.0	265.0	48.6	-	-	-	-
DINO [38]	1333	47.0	860.0	51.0	-	-	-	-
Swin-T [26]	-	86.0	745.0	50.5	-	-	-	-
Swin-S [26]	-	107.0	838.0	51.8	-	-	-	-
Swin-B [26]	1280	145.0	982.0	51.9	-	-	-	-
CNNs (real-time, without pre-trained)								
Yolov4 [1]	608	64.4	142.8	43.5	26.7	46.7	53.3	-
YoloX [7]	640	54.2	155.6	49.7	-	-	-	14.5
PPYoloE [37]	640	52.2	110.1	50.9	31.4	55.3	66.1	12.8
Yolov5 [14]	640	46.5	109.1	49.0	-	-	-	10.1
Yolov6 [16]	640	58.5	144.0	51.7	-	-	-	-
Yolov7 [35]	640	36.9	104.7	51.2	31.8	55.5	65.0	6.3
Yolov8 [15]	640	43.7	165.2	52.9	-	-	-	-
SYolo(ours)	640	57.2	142.5	**53.1**	**35.5**	**58.7**	**69.9**	10.3

Table 7: Architecture of the SYOLO. The E-ELAN/C2f is from the YOLOv7/8.

Stage	**Block**-l/s/n‡	**Channels**-l/s/n‡	**K. size**	**Stride**
Backbone				
P1	Conv	64 /16 /16	3	2
P2	ESD-I/II C2f ×3/2/1	128 /32 /32 128 /32 /32	3, 4 3, 1	2 1
P3	ESD-I/II C2f ×6/3/1	256 /64 /64 256 /64 /64	3, 4 3, 1	2 1
P4	ESD-I/II C2f ×6/3/1	512 /128 /128 512 /128 /128	3, 4 3, 1	2 1
P5	ESD-I/II C2f ×3/2/1 SPP ×1	1024 /256 /256 1024 /256 /256 1024 /256 /256	3, 4 3, 1 -	2 1 -
Neck(SA)				
P5-P4	SNI E-ELAN†×1	1536 /384 /284 2048 /512 /512	- 3, 1	- 1
P4-P3	SNI E-ELAN†×2/2/1	512 /192 /192 512 /256 /256	- 3, 1	- 1
P3-P4	ESD-I/II E-ELAN†×2/2/1	256 /128 /128 1024 /512 /512	3, 4 3, 1	2 1
P4-P5	ESD-I/II E-ELAN†×2/2/1	512 /256 /256 2048 /1024 /1024	3, 4 3, 1	2 1

† The first 3×3 vanilla convolutional layer of the E-ELAN is replaced by the GSConvE-I, and the end pointwise convolutional layer of the E-ELAN is replaced by the C2f.
‡ The 'l/s/n' means the model scale of the large/samll/nano.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 52172381), Graduate Research Innovation Foundation of Chongqing Jiaotong University (Grant No. CYB240259). The author would also like to thank Prof. Dr. Jun Li, Prof. Dr. Qiliang Ren, and Ms. Xinxin Liu (Intelligent Transportation Big Data Center, School of Information Science and Engineering, CQJTU) for providing GPUs.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
3. Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic detr: End-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2988–2997 (2021)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
5. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**, 98–136 (2015)
6. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
7. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
8. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
10. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1580–1589 (2020)
11. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
14. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Fang, J., Michael, K., Montes, D., Nadar, J., Skalski, P., et al.: ultralytics/yolov5: v6. 1-tensorrt, tensorflow edge tpu and openvino export and inference. Zenodo (2022)

15. Jocher, G., et al.: Yolo-ultralytics. https://github.com/ultralytics/ultralytics (2023)
16. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al.: Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
17. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13619–13627 (2022)
18. Li, H., Li, J., Wei, H., Liu, Z., Zhan, Z., Ren, Q.: Slim-neck by gsconv: a lightweight-design for real-time detector architectures. Journal of Real-Time Image Processing **21**(3), 62 (2024)
19. Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6054–6063 (2019)
20. Li, Z., Yang, L., Zhou, F.: Fssd: feature fusion single shot multibox detector. arXiv preprint arXiv:1712.00960 (2017)
21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
23. Liu, S., Huang, D., Wang, Y.: Learning spatial fusion for single-shot object detection. arxiv 2019. arXiv preprint arXiv:1911.09516 (1911)
24. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759–8768 (2018)
25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
27. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
28. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10213–10224 (2021)
29. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
30. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
31. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

33. Singh, B., Davis, L.S.: An analysis of scale invariance in object detection snip. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3578–3587 (2018)
34. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
35. 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. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7464–7475 (2023)
36. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10186–10195 (2020)
37. Yang, Z., Zhou, Y., Chen, Z., Ngiam, J.: 3d-man: 3d multi-frame attention network for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1863–1872 (2021)
38. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
39. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
40. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)