

Dynamic Scale Training for Object Detection

Yukang Chen^{1*}, Peizhen Zhang^{2*}, Zeming Li²,

Yanwei Li¹, Xiangyu Zhang², Lu Qi¹, Jian Sun², Jiaya Jia¹

¹ *The Chinese University of Hong Kong* ² *MEGVII Technology*

Abstract

We propose a Dynamic Scale Training paradigm (abbreviated as DST) to mitigate scale variation challenge in object detection. Previous strategies like image pyramid, multi-scale training, and their variants are aiming at preparing scale-invariant data for model optimization. However, the preparation procedure is unaware of the following optimization process that restricts their capability in handling the scale variation. Instead, in our paradigm, we use feedback information from the optimization process to dynamically guide the data preparation. The proposed method is surprisingly simple yet obtains significant gains (2%+ Average Precision on MS COCO dataset), outperforming previous methods. Experimental results demonstrate the efficacy of our proposed DST method towards scale variation handling. It could also generalize to various backbones, benchmarks, and other challenging downstream tasks like instance segmentation. It does not introduce inference overhead and could serve as a free lunch for general detection configurations. Besides, it also facilitates efficient training due to fast convergence. Code and models are available at github.com/yukang2017/Stitcher.

1. Introduction

Scale variation, a phenomenon that detection quality varies dramatically from one scale to another, originating from the imbalanced distribution of objects across different scales. It remains an unsolved challenge in object detection. In nature photography, it is impossible for an image to guarantee a balanced distribution of object patterns over different scales. Training model without handling this issue will not only depress the capability of detecting objects with minority scales but also hinder the overall performance.

Generally, existing methods alleviate the scale variance in virtue of *data preparation* or *model optimization*. For instance, in the data preparation literature, image pyramid [1] and multi-scale training augment inputs with multiple reso-

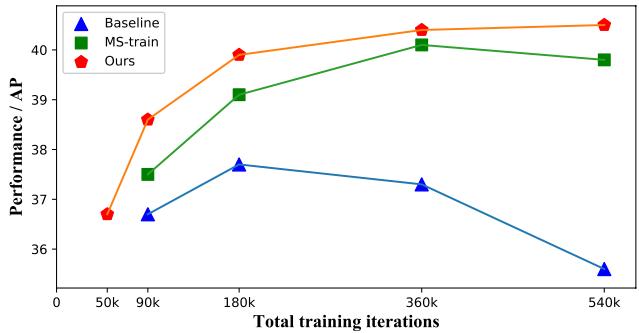


Figure 1. Performance varies from baseline and multi-scale training to ours as training proceeds. Experiments are conduct on Faster R-CNN [27] with ResNet-50 [13] FPN [17]. Our method consistently boosts the performance even for much longer training periods. However, the baseline and the multi-scale variant encounter severe over-fitting. Please refer to Table 5 for more details.

lutions. In model optimization, feature pyramids [17, 21, 8] enhance representations at different receptive levels. TridentNet and POD [16, 26] propose scale-invariant architectures for assembling dilated information. Ming *et al.* [24] design architectures by balanced loss penalization. However, the above methods omit the collaboration of data preparation and model optimization. On one aspect, the preparation strategies do not fully exploit the information from model optimization, and likely to produce static augmented data blind to the dynamic optimization requirements. Also shown in Figure 1, strategies like multi-scale training might encounter over-fitting if persistently learning the static data patterns¹. On another aspect, the model optimization tends to be sub-optimal if no training data with desired scale information is prepared. AutoAugment [5, 39] considers the collaboration during searching but their preparation strategy is static in the re-training stage. Besides, existing dynamic training methods focus on the collaboration to the label assignment, sample mining, or feature aggregation, without considering the data preparation.

In this paper, we propose a simple yet effective *Dynamic*

* Equal contribution. Work done during an internship in MEGVII.

¹Unlike [11] that using SyncBN [25] and GN [32], we fix BN [14] across all experiments for common settings.

Scale Training (DST) paradigm to mitigate the scale variation issue. This is accomplished by designing a feedback-driven, dynamic data preparation paradigm to meet the optimization requirement. To resolve the requirement, we opt for tracking the penalization intensities, instantiated by loss proportions over different scales. For convenience, we adopt the loss proportion owing to the minority scale of objects as feedback. Since these statistics reflect the scale variation information of the most underwhelming samples under the background of imbalanced optimization. We deem small scale to be the minority as is acknowledged. In general, the issues to concern are (1) how to devise a handy enough data preparation strategy with potential capability towards scale variation handling (2) how to dynamically guide this strategy, given loss proportion of small objects as feedback. For the first issue, we introduce a collage fashion of down-scaled images² (see Figure 3). This augmentation will potentially introduce objects with smaller sizes that might help rectify the optimization bias against majority scales (medium and large objects). Critically for the second issue, we devise a feedback-driven decision paradigm to dynamically determine the exploitation of the collage data, according to the loss statistics of the minority scales.

We experiment with our proposed DST method in various settings (backbones, training periods, and datasets). Results demonstrate that our method enhances performance consistently by handling scale variation. We also observe its versatility to different tasks by improving the performance on instance segmentation.

In summary, our contributions are two-fold:

- We propose a feedback-driven, dynamic data preparation paradigm for handling scale variation.
- We introduce a handy collage fashion of data augmentation, which would then be guided by the feedback at runtime.

2. Related Works

In this section, we shall give a brief retrospect to previous works about scale variation handling and start investigating the literature of dynamic training in object detection.

2.1. Scale Variation Handling

Current works for handling scale variation can be categorized into data preparation and model optimization.

Handling by Data Preparation Resampling is an intuitive method to handle scale variation, which is equivalent to amplify the loss magnitude of certain scales. However,

²other than direct re-scaling in multi-scale training that might cause extra overheads by potential large resolution of augmented data.

the improvement could be limited and might hurt the performance of the other scales (see Table 1). Image pyramid [1] has been popular since the era of hand-crafted descriptor learning to remedy scale variation. In recent years, multi-scale training becomes common for object detection. Features learned in this way are more robust to scale variation. However, both of the above strategies require additional overhead and storage consumption owing to transformed data with large resolutions. Moreover, since the target resolution is randomly chosen, an undesired data scale might be sub-optimal for handling scale variation.

SNIP and SNIPER [29, 30] are advanced versions of image pyramids. SNIP [29] is proposed to normalize the object scales under multi-scale training. SNIPER [30] sample patches, instead of regular inputs for training. It meticulously crops chips around the foregrounds and backgrounds to obtain training samples. However, the above methods rely on multi-scale testing that suffers from inference burden. Also, their strategies are fixed as training proceeds, overlooking the dynamic merits.

Unlike above specialized methods, customized augmentations like AutoAugment [5, 39] plausibly relieve the variation problem to some extent. These methods involve thousands of GPU days for optimizing the policy controller before actual re-training. Moreover, the searched policy is also fixed during re-training without adapting the optimization.

YOLOv4 [2], and Zhou et al. [37] involve similar image processing to our collage fashion. We claim the novelty about this since they are concurrent works to ours. YOLOv4 use Mosaic as data augmentation. Zhou, et al. crops foreground patches to construct jigsaw assembly for upstream classification. Instead, our method focuses on utilizing the collage images guided by dynamic feedback for handling scale variation.

Handling by Model Optimization Another line of effort for handling scale variation mainly exists in scale-invariant model optimization. This usually falls into two categories: the feature pyramids or the dilation based methods.

Feature pyramid methods aggregate information from multi-resolution levels. For instance, SSD [22] detects objects, taking as input the feature maps from different scales. Further, FPN [17] and its variants, e.g., PANet and NAS-FPN [21, 8] fully explore path aggregation to obtain high-level semantics across all scales. However, the aggregation manner is fixed during the model learning, without considering the adjustment for better training.

On the other hand, dilation based methods adaptively enlarge the receptive fields for scale robustness. Deformable convolution networks (DCN) [6] generalizes dilated convolution with flexible receptive regions. TridentNet [16] and POD [26] combine multiple branches with various dilation rates to extract scale-sensitive representations. However,

dilation based methods are not storage-friendly due to the high-resolution intermediate feature maps.

2.2. Dynamic Training for Object Detection

Currently, dynamic training utilized in object detection typically exists in online sample mining, feature aggregation, and label assignment. For sampling mining, OHEM [28] exploits region of interests (*RoIs*) for hard example mining according to the cost penalization. LapNet [4] introduces dynamic loss weight to indirectly conduct sample mining. For feature aggregation, FSAF [38] adaptively selects the most suitable features guided by the detection loss. ASFF [20] automatically learns the aggregation manner by dynamic masking. For label assignment, Liu *et al.* propose HAMBox [23] with dynamic compensation towards mismatched ground-truths. FreeAnchor [36] seeks for adaptive anchor-target matching during optimization. In MAL [15], the number of anchors shrinks progressively as the training proceeds. ATSS [35] proposes target-dependent training sample selection. Zhang *et al.* propose Dynamic R-CNN [34] for two-stage detectors. It progressively increases the Intersection-over-Union (*IoU*) threshold for better label assignment. However, none of the above methods refer to the data preparation which is also critical to the model training. In this paper, we propose an effective feedback-driven data preparation paradigm for scale variation handling.

3. Methodology

In this section, we shall briefly give a discussion about the scale variation issue. Subsequently, we will introduce the feedback-driven data preparation paradigm followed by the collage fashion of data augmentation. The overall pipeline of the proposed dynamic scale training framework is shown in Figure 2.

3.1. A Brief Discussion about Scale Variation

Scale variation refers to the phenomenon where models perform unfairly over different scales, featuring bad detection quality towards objects with minority scales. This commonly results from imbalanced frequencies of occurrence for instances belonging to different scales in the input images. Such imbalanced distribution would probably lead to biased network optimization. In many cases, the minority scales indicate the small scales.

Without loss of generality, we conduct statistics upon MS COCO [19] dataset and find two observations below:

- (a) *Imbalance across Dataset Does Not Affect*: Small³ objects hold above 41% instances in the dataset, breaking

³we follow the scale protocol in MS COCO [19] referred in Sec. 4.1. For fair annotation usage, we use the box area instead of the mask area as the size metric.

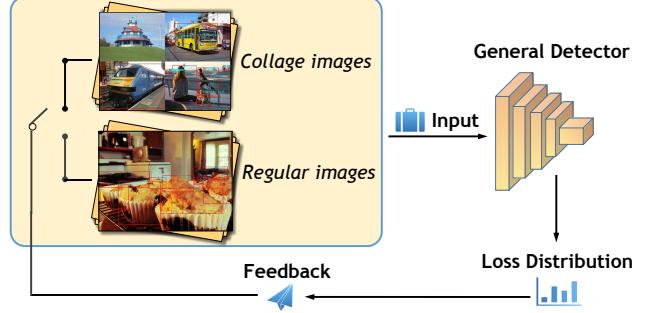


Figure 2. The pipeline of Dynamic Scale Training.

their rare stereotype. However, they still suffer from low-quality detection.

- (b) *Imbalance over Images that Matters*: medium and large objects exist in 71% and 83% of the images, respectively. In contrast, only around 52% of the images contain small objects.

Based on these observations, we believe that it is the imbalance over image distribution that leads to biased optimization towards different scales. This implies an overlooked concentration of the minority scales.

3.2. Our Approach

3.2.1 Feedback-Driven Data Preparation Paradigm

We propose a feedback-driven data preparation paradigm. In each training iteration, we fetch the loss proportion owing to small objects as feedback. It could be calculated after each forward propagation during model training.

Subsequently, if the loss proportion statistics is below a certain threshold in current iteration t , we deem it the timing to relieve imbalanced network optimization by latent compensation. In detail, we will construct collage images as input data instead of employing regular images in the next iteration $t+1$. Otherwise, if this statistic above the threshold, the regular images will serve as the input data in the coming iteration just like the default data preparation setting. The above binary deterministic paradigm could be summarized in Eq. (1) where I^{t+1} denotes the mini-batch data fed into the network at iteration $t+1$, I and I^c w.r.t. represent the regular and collage images in the coming iterations if applied. r_s^t denotes the loss proportion accounting for small-scale objects in iteration t . τ is the decision threshold to control data preparation.

$$I^{t+1} = \begin{cases} I^c, & \text{if } r_s^t \leq \tau, \\ I, & \text{otherwise.} \end{cases} \quad (1)$$

From another perspective, the proposed feedback-driven paradigm could be viewed as an agent optimization, ease of policy gradients in reinforcement learning. Specifically,

in the environment of object detector training, given the loss proportion observation in each training iteration, a non-parametric controller utilizes the aforementioned deterministic policy (specified in Eq. (1)) to sample from a binary action space, composed of regular or collage fashion of image processing for the next iteration of data preparation.

3.2.2 Collage Fashion of Data Augmentation

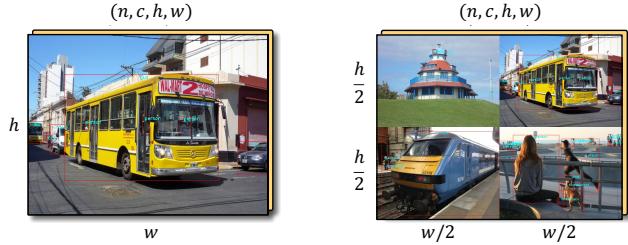


Figure 3. Regular images and collage images.

As stated in Sec. 1, we propose a collage fashion of scalable data augmentation for the purpose of convenient manipulation for dynamic training.

For simplicity and keeping the aspect ratio for retaining object shape priors, we formulate the collage by downscaling and stitching k regular images arranged in an equal number of rows and columns. Hence, k , equals to the square of row/column number, *e.g.*, 1, 2^2 , 3^2 , and so on. The spatial resolution of each component image inside is $(\frac{h}{\sqrt{k}}, \frac{w}{\sqrt{k}})$. Aside from the data, the box annotations of each source component image would get properly rescaled and translated for consistency. When k equal to 1, a collage image degenerates to a regular image.

Figure 3 shows a collage (right) specifying $k = 4$ compared to a regular image (left).

As can be seen, the collage fashion of image processing introduces a minimal scale variation handling by explicitly manufacturing object patterns with smaller scales. And Since collage images retain identical size as regular images, no additional overhead involves in network propagation.

4. Experiments

In this section, we begin by briefly describing the implementation details. Whereafter, the efficacy analysis of our proposed method compared to previous works are investigated. Next, we shall elaborate on the ablation studies. A quantitative analysis of the scale variation issue will also be given. We end the experiment section by discussing extra merits and corner cases brought by the proposed method.

4.1. Implementation Details

Experiments are mainly conducted on the challenging MS COCO [19] dataset which contains 80 categories. Fol-

lowing the common practice in [10], the union of the primitive training set (80k images) and the `trainval35k` subset (35k images) of primitive validation set are used for training. The evaluation is conducted on the `minival` subset with 5k images. We follow the scale protocol in COCO to distinguish the small, middle and large objects by 32^2 and 96^2 pixel areas. Input images are resized such that their shorter side is 800 and the longer side no more than 1,333.

Throughout all experiments, the initial learning rate is set as 0.02 with Stochastic Gradient Descent (SGD) with momentum as 0.9 and weight decay as 1e-4. The mini-batch size is set to 16 (2 images per GPU). The network is trained for 90k iterations that will be decayed by 10 at 60k and 80k iterations, respectively. For longer training periods if required, we adopt a common proportional milestones extension. For example, a $2\times$ setting with 180k iterations, and milestones at 120k and 160k respectively.

Besides MS COCO, we also examine our efficacy of handling scale variation on PASCAL VOC [7] dataset. Moreover, extra studies on the challenging instance segmentation task also verify the versatility of our proposed method.

4.2. Comparison to Previous Methods

4.2.1 Comparison to Resampling

Following the spirit of the resampling strategy for more balanced training, we apply a careful re-weight scheme to assist the minority scales in each iteration. In detail, we amplify the loss magnitudes of small objects to be equal to that of the medium and large objects. However, as shown in Table 1, the overall performance and the performance of the other scales deteriorate with an only slight improvement to the small scale ($AP_s + 0.3\%$).

Table 1. Impact brought by resampling.

	AP	AP_s	AP_m	AP_l
Baseline	36.7	21.1	39.9	48.1
+ Resampling	36.4	21.4	39.3	47.4

4.2.2 Comparison to Common Baselines

As shown in Table 2, the improvement against baseline is highlighted in parenthesis. We observe decent improvement overall ($1.7\% + AP$), and more significant results for the minority scales, *i.e.*, the small scales ($3.2\% + AP_s$). Table 3 shows the comparison in $2\times$ training periods, presenting even higher gains ($2.2\% + AP$ and up to $4.0\% AP_s$).

We also conduct counterpart experiments on single stage detectors, *e.g.*, RetinaNet [18] and FCOS [31] as shown in Table 6.

These demonstrate the effectiveness not only on general detection enhancing but also on scale variation handling *esp.* for the minority scales using dynamic scale training.

Table 2. Comparison with common baselines and multi-scale training on Faster R-CNN.

	Backbone	Hours	AP	AP _s	AP _m	AP _l
Baseline	ResNet-50 FPN	8.7	36.7	21.1	39.9	48.1
MS-train ^s		8.1	36.3	23.7 (+2.6)	39.9	45.9 (-2.2)
MS-train ^m		10.8	37.5	22.0	40.7	48.8
MS-train ^l		14.4	37.1	20.7 (-0.4)	40.3	49.8 (+1.7)
Ours		9.0	38.6 (+1.9)	24.4 (+3.3)	41.9 (+2.0)	49.3 (+1.2)
Baseline	ResNet-101 FPN	11.5	39.1	22.6	42.9	51.4
MS-train ^s		10.8	38.9	24.2 (+1.6)	42.7	49.0 (-2.4)
MS-train ^m		14.2	39.7	23.6	43.3	51.3
MS-train ^l		21.3	39.3	22.3 (-0.3)	43.0	51.9 (+0.5)
Ours		11.7	40.8 (+1.7)	25.8 (+3.2)	44.1 (+1.2)	51.9 (+0.5)

Table 3. Comparison with common baselines and multi-scale training on Faster R-CNN for 2x training periods.

	Backbone	Hours	AP	AP _s	AP _m	AP _l
Baseline	ResNet-50 FPN	17.2	37.7	21.6	40.6	49.6
MS-train ^m		20.5	39.1	23.5	42.2	50.8
Ours		17.5	39.9 (+2.2)	25.1 (+3.5)	43.1 (+2.5)	51.0 (+1.4)
Baseline	ResNet-101 FPN	23.4	39.8	22.9	43.3	52.6
MS-train ^m		28.5	41.6	25.5	45.3	54.1
Ours		23.5	42.1 (+2.3)	26.9 (+4.0)	45.5 (+2.2)	54.1 (+1.5)

4.2.3 Comparison to Multi-scale Training

(a) Different settings of multi-scale training

We carefully compare our method with multi-scale training with various scale settings as exhibited in Table 2. Here, MS-train^s, MS-train^m and MS-train^l correspond to sampling intervals about the shorter side length, denoted as [400, 800], [600, 1000], and [800, 1200] respectively, with stride 100. They indicate settings prefer to small, middle, and large scale respectively. Among them, Multi-scale^m achieves the best trade-off, as the other two settings acquire improvement in their favorite scale at the price of greatly harming the opposite scale (highlighted in blue and green in Table 2). Hence, We adopt the Multi-scale^m setting for Multi-scale training in the following experiments. Yet, our method still outperforms this strategy across all scales.

(b) Time efficiency

The proposed dynamic scale training method brings about negligible overhead compared to baselines. It mainly comes from the collage augmentation which involves *nearest* neighbor interpolation for down-scaling component images. Empirically, a collage operation costs about 0.02 seconds in a single training iteration. Since the frequencies of collage operation depend on the dynamic preparation paradigm that is unavailable in advance. We practically measure the time consumption in terms of the complete training period. All measurements are benchmarked on 8 RTX 2080Ti GPU cards with 16 mini-batch size.

As shown in Table 2, it takes 8.7 hours to train the baseline with ResNet-50 FPN in 1× period. Instead, multi-scale training requires extra 2 hours (10.8). The gap enlarges when experimenting on a larger backbone (ResNet-

101 FPN) or longer training period (2×). In contrast, our method takes only a bit longer than the baseline (9 hours with extra 0.3 hours). And the gap is invariant to the training periods (nearly the same in both 1× and 2× settings). Moreover, the gap shrinks when taking larger backbones (ResNet-101 FPN) for experiments. Please refer to Table 2 and Table 3 for details. Therefore, our proposed method is much more efficient than multi-scale training.

Table 4. Evaluation on the effect of multi-scale testing.

	AP	AP _s	AP _m	AP _l
MS-train ^m	37.5	22.0	40.7	48.8
+ MS-test ^m	38.8 (+1.3)	23.7	41.6	49.8
Ours	38.6 39.9 (+1.3)	24.4	41.9	49.3
+ MS-test ^m		26.5	42.7	51.0

(c) Compatible to multi-scale testing

It is acknowledged that models trained with multi-scale training could further enhance the performance with matching multi-scale testing. Thus, without loss of generality, we conduct a comparison by applying MS-test^m to MS-train^m and our proposed method, respectively. As shown in Table 4, our proposed method shares exactly the same merit (+1.3%). This reveals good compatibility.

(d) Longer training periods

Recalling the proposed collage augmentation, one association with multi-scale training is that they both create scalable instance patterns to some extent. However, we are wondering if multi-scale training is capable of mitigating the performance gap to ours, if long enough training periods are allowed.

Table 5. Evaluation on longer training periods.

	Iterations	AP	AP _s	AP _m	AP _l
Baseline	90k	36.7	21.1	39.8	48.1
	180k	37.7	21.6	40.6	49.6
	360k	37.3 ↓	20.3	39.6	50.1
	540k	35.6 ↓	19.8	37.7	47.6
MS-train	90k	37.5	22.0	40.7	48.8
	180k	39.1	23.5	42.2	50.8
	360k	40.1	24.3	43.3	52.4
	540k	39.8 ↓	24.1	43.0	52.0
Ours	90k	38.6	24.4	41.9	49.3
	180k	39.9	25.1	43.1	51.0
	360k	40.4	25.2	43.6	51.9
	540k	40.5 ↑	26.1	43.2	51.6

To resolve this, we conduct experiments upon Faster R-CNN with ResNet-50 and FPN on various training periods as shown in Table 5. We find that the gap starts shrinking when the training process reaches sufficiently longer periods ($3\times$ to $4\times$). However, interestingly for the longest $6\times$ training period (540k iterations), the performance of the multi-scale training (also the baseline) encounter degradation. Instead, our method could further enhance the performance. One reasonable explanation is that the feedback-driven preparation paradigm consistently provides data of the desired scale to effectively avoid *over-fitting*.

4.2.4 Comparison to SNIP and SNIPER

As shown in Table 7, we compare our method to SNIP [29] and SNIPER [30]⁴ methods on various backbones. As a result, our method performs better. This might because the SNIP and SNIPER operate in a static manner during training, rendering them unable to provide scale-sensitive data that the network desires. In contrast, our method benefits from the dynamic data preparation paradigm to meet the requirements training-dependently. Moreover, our method is simpler to use while SNIPER involves extended label assignment and chips sampling procedure.

4.2.5 Evaluation on Large Backbones

Table 8 shows the improvement from our method on large backbones, *i.e.*, ResNext 101 [33], ResNet-101 with DCN [6] and ResNext-32×8d-101 with DCN [6]. Based on the strong baselines, our method could still enhance the performance by 1.0% to 1.5% AP.

⁴For fair comparisons, we use the same augmentations (deformable convolution, MS test, and soft-NMS [3]) as SNIP and SNIPER do.

4.2.6 Evaluation on Instance Segmentation

Beyond object detection, we also apply our method to instance segmentation task. Experiments are conducted on the COCO instance segmentation track [19]. We report COCO mask AP on the minival split. Models are trained for 90k iterations and divided by 10 at 60k and 80k iterations. We train Mask R-CNN [12] models with Stochastic Gradient Descent (SGD), 0.9 momentum and 1e-4 weight decay and 16 batch size (2 images for per GPU). As shown in Table 9, our method improves AP by 0.9% on ResNet-50 and by 1.3% on ResNet-101.

4.2.7 Evaluation on PASCAL VOC

Besides MS COCO, we also generalize our proposed dynamic scale training method to Pascal VOC [7] dataset. Following the protocol in [9], the union of 2007 trainval and 2012 trainval are used for training. Models are trained by 24k iterations in which the learning rate is set as 0.01 and 0.001 in the first two-thirds and the remaining one-third iterations, respectively. Evaluation is performed on 2007 test. As shown in Table 10, our method obtains a gain of 2.3% mAP overall. In addition, the detection quality of small scale categories like bottle, chair, and tv get significantly improved.

4.3. Ablation Studies

In this section, we analyze the best practice of the feedback choice and the deterministic threshold τ in the feedback-driven data preparation paradigm. Besides, we conduct a simple ablation on selecting the number of component images k in the collage fashion. We use Faster R-CNN with ResNet-50 and FPN for the studies.

Feedback choice. To explore the preparation paradigm, we set up below control experiments as shown in Table 11.

- *All collage*: collage images all the time;
- *All regular*: regular images all the time (baseline);
- *Random sampling*: collage or regular images randomly;
- *Input feedback*: occurrence frequency of small instances in the input as feedback;
- *Classification/Regression/Joint loss feedback*: loss proportion of small objects as feedback.

As shown in Table 11, static usage of collage images leads to bad performance. It might run into another extreme situation where learning biases towards small-scales. Besides, random sampling performs better than the common baseline, but it is still static. The dynamic feedback strategies, *e.g.*, the input feedback, results in better performance. However, such input-guided feedback is inferior to the loss-guided ones since it does not consider the optimization process. Results with different loss-guided feedback strategies

Table 6. Comparison on RetinaNet and FCOS with ResNet-50 and ResNet-101 backbones for $2\times$ training periods.

	Model	Backbone	AP	AP_s	AP_m	AP_l
Baseline	RetinaNet	ResNet-50 FPN	36.8	20.2	40.0	49.7
Ours			39.0 (+2.2)	23.4 (+3.2)	42.9 (+2.9)	51.0 (+1.2)
Baseline	FCOS	ResNet-101 FPN	38.8	21.1	42.1	52.4
Ours			41.3 (+2.5)	25.4 (+4.3)	45.1 (+3.0)	54.0 (+1.6)
Baseline	FCOS	ResNet-50 FPN	37.1	21.6	41.0	47.3
Ours			39.8 (+2.7)	25.4 (+3.8)	43.9 (+2.9)	50.2 (+2.9)
Baseline	FCOS	ResNet-101 FPN	39.1	22.2	43.4	50.6
Ours			41.6 (+2.5)	26.1 (+3.9)	45.5 (+2.1)	53.3 (+2.7)

Table 7. Comparison with SNIP / SNIPER.

	Backbone	AP	AP_s	AP_m	AP_l
SNIP	ResNet-50 C4	43.6	26.4	46.5	55.8
SNIPER		43.5	26.1	46.3	56.0
Ours		44.2	28.7	47.2	58.3
SNIP	ResNet-101 C4	44.4	27.3	47.4	56.9
SNIPER		46.1	29.6	48.9	58.1
Ours		46.9	30.9	50.5	60.9

Table 8. Evaluation on Large Backbones.

	Backbone	AP	AP_s	AP_m	AP_l
Baseline	ResNext 101	41.6	24.8	45.1	53.3
Ours		43.1	28.0	46.7	54.2
Baseline	ResNet 101 + DCN	42.3	24.8	46.1	55.7
Ours		43.3	27.1	47.0	56.0
Baseline	ResNext 101 + DCN	44.1	26.8	47.5	57.8
Ours		45.4	29.4	48.8	58.5

Table 9. Evaluation on Instance Segmentation.

	Backbone	AP	AP_s	AP_m	AP_l
Baseline	ResNet-50 FPN	34.3	15.8	36.7	50.5
Ours		35.1	17.0	37.8	51.4
Baseline	ResNet-101 FPN	35.9	15.9	38.9	53.2
Ours		37.2	19.0	40.3	53.7

are comparable, robust to specific supervision tasks. By default, we use regression loss-guided for convenience.

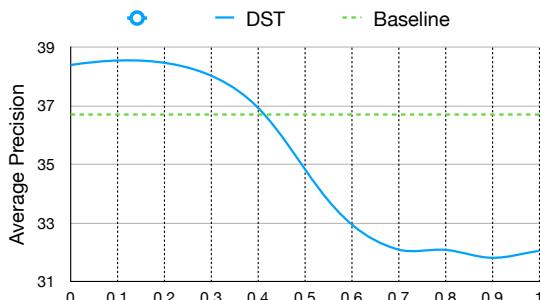


Figure 4. Ablation study on the threshold τ .

Deterministic threshold. In the proposed method, only one hyper-parameter τ requires tuning. We apply grid searching and study the impact as shown in Figure 4. The performance decreases dramatically as τ exceeds 0.2. Empirically, we set τ as 0.1 and apply it across all experiments without loss of generality. Notably, it happens to be coincident with the ratio observation covering half of the training iterations, as described in Figure 5. This provides a promising heuristics for convenient tuning by first calculating the statistics during the baseline training on a minimal subset.

Number of collage components. We conduct a simple ablation on different number k of component images used in collage by our proposed method. Since we mainly focus on the dynamic preparation paradigm, we simply adopt $k = 4$ for a good trade-off as shown in Table 12.

4.4. Analysis of Scale Variation

Besides reflecting the improvement of scale variation handling by performance gains, we also investigate in the view of optimization preference. We measure this by loss proportions occupied by different scales over iterations. These statistics are collected from the training process of Faster R-CNN with ResNet-50 and FPN. As a result, we draw the curves of model training w/ and w/o our proposed method in Figure 5. It can be observed that the scale variation gets much alleviated.

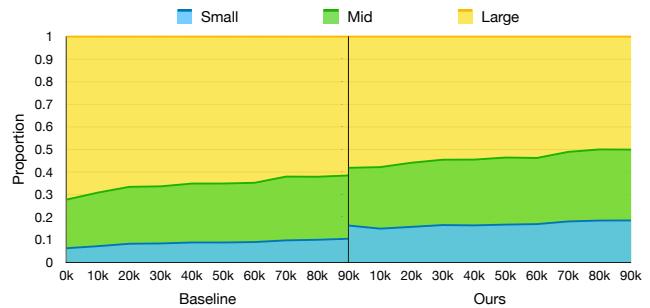


Figure 5. Loss proportion across different scales before and after.

Beyond the overall observation, we also investigate into loss proportions of the small scales. As shown in the Figure 6 left, more than half of the training iterations undergo

Table 10. Evaluation on PASCAL VOC dataset on Faster R-CNN.

	mAP	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv
Baseline	80.3	86.9	86.7	80.1	72.5	71.9	86.9	88.4	88.7	63.3	87.0	75.3	88.5	88.4	80.1	85.5	56.7	78.2	78.8	85.0	77.6
Ours	82.6	89.0	86.7	80.2	73.0	72.7	87.0	89.3	89.0	68.6	86.8	79.7	88.8	88.5	88.1	87.3	59.8	86.7	80.2	88.1	84.0

Table 11. Ablation study on feedback choice.

feedback strategy (if any)		AP	AP _s	AP _m	AP _l
No	All collaged	32.1	21.9	36.4	36.8
	All regular	36.7	21.1	39.8	48.1
	Random sampling	37.8	23.6	40.7	46.7
Yes	Input Ratio	38.1	23.1	41.3	49.1
	Classification Loss	38.5	23.9	41.6	48.8
	Regression Loss	38.6	24.4	41.9	49.3
	Joint Loss	38.5	23.7	41.6	49.3

Table 12. Ablation study on number of collage components.

k	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
1 ²	36.7	58.4	39.6	21.1	39.8	48.1
2 ²	38.6	60.5	41.8	24.4	41.9	49.3
3 ²	38.4	60.5	41.5	24.2	41.7	48.8

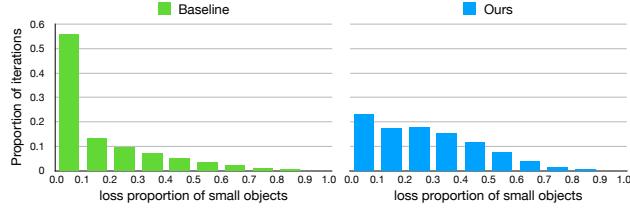


Figure 6. Loss proportion of minority scales before and after.

an extremely low loss proportion of small objects, dubbed 0.1. By adopting our proposed method, the scale variation from the perspective of loss proportion distribution get much alleviated (see Figure 6 right).

4.5. Other Merits

Beyond the performance enhancement, there are also extra merits brought by the proposed dynamic scale training during scale variation handling.

4.5.1 Speed-Accuracy Trade-off

We find an improvement upon speed-accuracy trade-off as shown in Table 13. It could be observed that our method runs on par with the baseline (AP: 37.0 vs. 36.7) given inputs of much smaller sizes (resolution: (512, 853) vs. (800, 1333)) and meanwhile is 1.6× faster.

4.5.2 Fast Convergence

We discover the fast convergence capacity of our proposed DST method. Referring to Table 14, after applying DST, it

Table 13. Speed-accuracy trade-off merit brought by dynamic scale training. The baseline is Faster R-CNN with ResNet-50 and FPN.

	Resolution	Inference time	AP
Baseline	(800, 1333)	56 ms / img	36.7
Baseline	(512, 853)	35 ms / img	33.5
Ours	(800, 1333)	56 ms / img	38.6
Ours	(512, 853)	35 ms / img	37.0

Table 14. Fast convergence merit brought dynamic scale training. The baseline is Faster R-CNN with ResNet-50 and FPN.

	Iterations	AP	AP _s	AP _m	AP _l
Baseline	90k	36.7	21.1	39.9	48.1
Ours	50k	36.7	22.9	39.9	46.6
Ours	90k	38.6	24.4	41.9	49.3

nearly halves (iters: 50k vs. 90k) the training iterations to achieve the same accuracy to the baseline.

4.6. Corner cases of collage

Recalling the collage procedure, regular images are down-scaled before being stitched to form the collage components. This might produce extremely tiny objects more likely to be a noisy pattern (from existing small objects). To investigate the impact, we discard tiny samples whose box areas less than 100 pixels. Before removal, the results are AP: 38.6, AP_s: 24.4, AP_m: 41.9, AP_l: 49.3. After removal, we obtain AP: 38.6, AP_s: 24.7, AP_m: 41.8, AP_l: 49.1. This demonstrates that tiny patterns do not affect the overall performance but might hamper the quality on small scales.

5. Conclusion

In this paper, we propose a simple yet effective *dynamic scale training method (DST)* for object detection. By relieving the scale variation issue in virtue of feedback information from the optimization process, we observe significant gains in detection performance. Moreover, it introduces efficient convergence during training and does not affect the inference time as a free lunch. Abundant experiments have been conducted to verify its efficacy on various backbones, training periods, datasets, and different tasks. DST could be easily incorporated into modern detectors and steadily enhances the detection quality. We expect it could serve as a common configuration in the future, facilitating further dynamic training research for object detection.

References

- [1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. [1](#) [2](#)
- [2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020. [2](#)
- [3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms - improving object detection with one line of code. In *ICCV*, pages 5562–5570, 2017. [6](#)
- [4] Florian Chabot, Mohamed Chaouch, and Quoc Cuong Pham. Lapnet: Automatic balanced loss and optimal assignment for real-time dense object detection. *CoRR*, abs/1911.01149, 2019. [3](#)
- [5] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, pages 113–123, 2019. [1](#) [2](#)
- [6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. [2](#) [6](#)
- [7] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. [4](#) [6](#)
- [8] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, pages 7036–7045, 2019. [1](#) [2](#)
- [9] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015. [6](#)
- [10] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. [4](#)
- [11] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, pages 4918–4927, 2019. [1](#)
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. [6](#)
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [1](#)
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. [1](#)
- [15] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *CVPR*, pages 10206–10215, 2020. [3](#)
- [16] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. 2019. [1](#) [2](#)
- [17] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017. [1](#) [2](#)
- [18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017. [4](#)
- [19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. [3](#) [4](#) [6](#)
- [20] Songtao Liu, Di Huang, and Yunhong Wang. Learning spatial fusion for single-shot object detection. *CoRR*, abs/1911.09516, 2019. [3](#)
- [21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. [1](#) [2](#)
- [22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016. [2](#)
- [23] Yang Liu, Xu Tang, Xiang Wu, Junyu Han, Jingtuo Liu, and Errui Ding. Hambox: Delving into online high-quality anchors mining for detecting outer faces. *CoRR*, abs/1912.09231, 2019. [3](#)
- [24] Xiang Ming, Fangyun Wei, Ting Zhang, Dong Chen, and Fang Wen. Group sampling for scale invariant face detection. In *CVPR*, pages 3446–3456, 2019. [1](#)
- [25] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *CVPR*, pages 6181–6189, 2018. [1](#)
- [26] Junran Peng, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Pod: practical object detection with scale-sensitive network. In *CVPR*, pages 9607–9616, 2019. [1](#) [2](#)
- [27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. [1](#)
- [28] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016. [3](#)
- [29] Bharat Singh and Larry S. Davis. An analysis of scale invariance in object detection SNIP. In *CVPR*, pages 3578–3587, 2018. [2](#) [6](#)
- [30] Bharat Singh, Mahyar Najibi, and Larry S. Davis. SNIPER: efficient multi-scale training. In *NeurIPS*, pages 9333–9343, 2018. [2](#) [6](#)
- [31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *CVPR*, pages 9627–9636, 2019. [4](#)
- [32] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, pages 3–19, 2018. [1](#)
- [33] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. [6](#)
- [34] Hongkai Zhang, Hong Change, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *ECCV*, 2020. [3](#)

- [35] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9759–9768, 2020. [3](#)
- [36] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *NIPS*, pages 147–155, 2019. [3](#)
- [37] Dongzhan Zhou, Xinchi Zhou, Hongwen Zhang, Shuai Yi, and Wanli Ouyang. Cheaper pre-training lunch: An efficient paradigm for object detection. In *ECCV*, 2020. [2](#)
- [38] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, pages 840–849, 2019. [3](#)
- [39] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In *CVPR*, page 770–778, 2019. [1](#), [2](#)