Abstract

Scale variation remains a challenge problem for object detection. Common paradigms usually adopt multi-scale training & testing (image pyramid) or FPN (feature pyramid network) to process objects in wide scale range. However, multi-scale methods aggravate more variation of scale that even deep convolution neural networks with FPN cannot handle well. In this work, we propose an innovative paradigm called Consistent Scale Normalization (CSN) to resolve above problem. CSN compresses the scale space of objects into a consistent range (CSN range), in both training and testing phase. This reassures problem of scale variation fundamentally, and reduces the difficulty for network learning. Experiments show that CSN surpasses multi-scale counterpart significantly for object detection, instance segmentation and multi-task human pose estimation, on several architectures. On COCO test-dev, our single model based on CSN achieves 46.5 mAP with a ResNet-101 backbone, which is among the state-of-the-art (SOTA) candidates for object detection.

1. Introduction

The vision community has rapidly improved performance on object recognition, especially object detection [24], instance segmentation [11], human pose estimation [11]. Among them, Object detection is a prerequisite for many downstream applications. The performance of detectors has dramatically make progress with the help of powerful backbone networks [12], delicate design on optimization objective [16] and well-annotated datasets [17].

However, detecting objects of various scales remains challenging, especially encountering objects of extreme size. As shown in Table 1, AP on small objects falls much compared to medium and large objects. To alleviate the scale variation problem, state-of-the-art detectors rely on feature pyramids [15] or image pyramids[5]. On the one hand of feature pyramids, FPN [15] construct a multi-stage network with parallel prediction on objects of isolated scale range. On the other hand of image pyramids, simple multi-scale training & testing strategy still play a role on multiple recognition tasks. In particular, [26, 27, 20] found that ignoring loss signals from extremely tiny and large objects can improve detection accuracy.

It should be pointed out that aforementioned methods have defects respectively. According to [30], the receptive field and semantic scope of the same RoI should be consistent in feature map. For object of large scale, the receptive field may not be sufficient; while for the small one, the semantic scope is larger comparing to object’s size. So there exists consistency only when object’s size falls into a moderate range. Multi-scale training resize images to different resolutions, therefore resizing some objects to normal scales. But it also let some objects to extreme scales with great inconsistency, which give rise to final accuracy degradation. As shown in Table 2, the detector trained from image pyramid with wider scale range is inferior to the one from normal scale range. FPN [15] employed a heuristic rule to make feature map on different stage responsible for RoIs of isolated scale range. It also suffer from the inconsistency when encountering tiny objects (downsampled to sub-pixels and hard to recognize) or large objects (receptive field cannot cover). SNIP [27] tried to ignoring the extremely tiny and large objects, resulting removing the inconsistency on these extreme samples. But there also exists inconsistency about the scale range usage. The detailed analysis will be reported in 3.2.

We propose a simple and effective method approach, called Consistent Scale Normalization (CSN), to further alleviate the inconsistency resulting from scale variation. CSN will also train detector on image pyramid, but only optimize model on objects in moderate scale range. With FPN integrated, the heuristic rule in FPN can distribute RoIs of large scale range to multiple stage which process RoIs in smaller range. This alleviates the learning difficulty and enlarges feasible range on object scales, resulting more avail-
Table 1: Results of different scales on Faster R-CNN of ResNet-50

| FPN  | AP   | AP$_S$ | AP$_M$ | AP$_L$ |
|------|------|--------|--------|--------|
|      | 34.7 | 16.3   | 38.2   | 49.3   |
| ✓    | 38.0 | 22.0   | 41.5   | 48.8   |

Table 2: We train two detectors using image scales randomly from a normal range [640, 800] and larger range [160, 1600] respectively. Both detectors are tested on single scale 800. The accuracy from larger scale range is worse than the normal one. Detectors are simple Faster R-CNN on ResNet-18.

Table: We train two detectors using image scales randomly from a normal range [640, 800] and larger range [160, 1600] respectively. Both detectors are tested on single scale 800. The accuracy from larger scale range is worse than the normal one. Detectors are simple Faster R-CNN on ResNet-18.

2. Related Works

Driven by representation capacity of deep conv-feature, CNN-based detectors [7, 24, 18, 22] are a dominant paradigm in object detection community. The R-CNN series and variants [24, 6, 24] gradually promotes the upper bound of performance on two-stage detectors. In particular, Faster R-CNN [24] adopts shared backbone network to proposal generation and RoI classification, resulting real-time detection and accuracy rising. Mask R-CNN [11] introduced a multitask framework for related recognition tasks, such as instance segmentation and human pose estimation, achieving higher accuracy with brief procedure and structure. On the other hand, YOLO-v3 [23], RetinaNet [16], CornerNet [13] and etc, rely on only one stage to predict, while struggling to catch up with the top two-stage detectors. To compare and verify effectiveness of proposed method, CSN model is implemented based on Faster R-CNN [24] and Mask R-CNN [11] for various tasks.

There still exists difficulties for detectors on objects with large scale variation. Early solutions choose learning scale invariant representation for object to tackle scale variation problem. For traditional members, Haar face detector [28, 29] and DPM [5] become more scale-robust with the help of image pyramid [1]. SSD [18], SSH [19], MS-CNN [3] try to detect small objects at lower layers, while big objects at higher layers. To be further, FPN [15] fuses feature maps at adjacent scale to combine semantics from upper and details from lower. Objects of different size can be predicted at corresponding levels according to a heuristic rule. SAFD[10] predicts scale distribution histogram of face, which guides zoom-in and zoom-out for face detection. FSAF[32] selected the most suitable level of feature for each object dynamically in training compared with FPN. SNIP [26] assumes it’s easy for detectors to generalize well to objects of moderate size. Only objects of normal scale range are utilized to make model converge better. Ultra-.orly, SNIPER[27] effectively mined on generated chips for better result.

Compared to SNIP etc [26, 27], CSN adopts a consistent scale normalization strategy to select training samples, and integrates FPN to attain better generalization. It also extends to other recognition tasks, validating the effectiveness of our method.

3. Method

Current models still suffer from large scale variation and cannot obtain satisfactory accuracy even with multi-scale training & testing. We will introduce our consistent scale normalization(CSN) in this section to deal with it better. Concretely, in section 3.1, the common Faster R-CNN detector is recapped. In section 3.2, we analyze the drawbacks of SNIP which motivates the proposal of CSN. In section 3.3, we detail the object sampling mechanism of CSN technique, including the effects on FPN. In section 3.4, details of CSN on various recognition tasks are described.

3.1. Faster R-CNN detector recap

Faster R-CNN and its variants are leading detectors in object detection community, currently. They basically consists of two steps. In the 1st-stage, region proposal network (RPN) generates a bunch of RoIs (Region of Interest)
on basis of a set of pre-defined anchors. These RoIs indicate the region where possible objects exist in the image. Then, in the 2nd-stage, a Fast R-CNN[6] extracts fixed size feature (e.g., $7 \times 7$) for each RoI from the shared feature with RPN. This can be implemented by RoIPooling [24] or RoIAlign [11] operators. Finally, these features will be sent to two independent subnets for category classification and box regression, respectively.

In Faster R-CNN, all ground-truth object bounding-boxes (gt-bboxes) in current image are collected to participate in training.

### 3.2. Object scale range

We think that ConvNets intrinsically suffer from large scale variation in training, as shown in Table 1. An ordinary solution is diminishing the large variation, namely sampling objects in moderate scale range. SNIP [26] gives a detailed analysis on effect of object-scale to training. However, we argue that there exists inconsistency in SNIP’s object selection mechanism.

#### 3.2.1 Inconsistency in SNIP

SNIP is a training scheme designed for multi-scale training, e.g., with image pyramid. It aims at excluding objects of extreme size in training phase. In details, the valid training range in original image is carefully tuned for each image resolution. Then for the $i$-th resolution, the RoIs whose area (w.r.t original image) fall in the valid range $[s_i^c, e_i^c]$ will participate in training. Otherwise, they are ignored.

However, there is a basically unreasonable case with this training paradigm. That is, the objects with nearly same scale in resized images may not take part in training together. As illustrated in Fig 1, the man on the left in Fig 1(b) and woman player on the leftmost in Fig 1(d) shares same scale in resized resolutions. The former can take part in training while the latter cannot.

Fig 1(c) gives the distribution of training objects in SNIP. As it shows, there are plentiful extremely tiny and huge objects participating in training phase. The Fig 1(f) exhibits the distribution of ignored objects in SNIP. As it is shown, the ignored objects overlaps much with the trained objects, which is the cause of the contradiction in previous unreasonable case. In a word, the ignoring mechanism is implicit and inconsistent in SNIP. So, the model’s behavior is uncertain and the most suitable scale of the model for testing is indeterminate. In SNIP [26], the valid range for testing is obtained via greedy search in val-set.

So we argue the valid range here is not consistent between training and testing phase, as well as among different resolutions. This results in burdensome tuning of hyper-parameters. For example, for a three-level image pyramid, one need to tune three pairs of valid ranges for training and another three pairs for testing. The inconsistency could also hamper the model performance due to self-contradictory sampling strategy.

### 3.3. Consistent Scale Normalization

Previous scale normalization method, e.g., SNIP has some self-contradiction in its sampling strategy. Here, consistent scale normalization (CSN) method is an improved multi-scale training & testing strategy with consistent scale normalization on objects.
Figure 2: **CSN plus FPN**: Here shows CSN with 3 scaling factors for multi-scale training & testing. In training, CSN firstly resizes original image \( \omega_i \) times to get \( i \)-th resolution. Then it select objects which scale falls in CSN range (marked by green boxes) as valid for training, otherwise discards invalid boxes (red). In testing, only the predicted boxes from different resolutions in CSN range are preserved and fused.

### 3.3.1 Consistent scale normalization

With the same chip generation strategy in SNIPER [27], CSN employs the scale normalization that adopts consistent scale range \([s_l, s_u]\), called CSN range, for different phases and resolutions. In training phase, firstly, scaling factor set \( \Omega \) is pre-defined. For the original image with size \([h, w]\), it is resized to \( i \)-th resolution \([h \cdot \omega_i, w \cdot \omega_i]\) by \( \omega_i \in \Omega \). Then each RoI whose scale on \( i \)-th resolution falls in this range will be set to valid currently, else invalid, as shown in Fig 2. The invalid objects constitute the ignored regions which don’t contribute to training. In testing phase for each resized resolution, all predicted RoIs whose scale falls within the same CSN range will be kept for post-processing. In this way, all objects can be normalized to a moderate scale range consistently. This improved selection strategy is obviously more excellent. On the one hand, uniform scale range eliminates the adaption of scale between training and testing phase. On the other hand, the consistent scale normalization reduces scale fluctuation of objects trained on different image resolutions. What’s more, CSN only tune the two parameters, i.e., CSN range \([s_l, s_u]\), no matter how many image resolutions are used, while SNIP is in linear with that. For example, for 3-level image pyramid, the number of scale range parameters is 12 with SNIP.

### 3.3.2 Feature pyramid integration

When setting scale range limitation for objects, the number of valid objects will decrease, resulting in possible over-fitting, as [14] shows. Even taking multi-scale training method, this over-fitting influence cannot be eased.

FPN adopts separate stage, i.e., \( P_2, P_3, P_4, P_5 \), to predict objects at disjoint scale range, and has better capability for detecting objects in wide scale range. Even if enlarging the CSN scale range when integrating FPN, each stage can also be trained well because of the reduced range. CSN turns to feature pyramid network (FPN) for help. Not only does FPN bring more powerful feature representation for objects, but also enlarge the feasible scale range.

The experiments also verifies positive effect of CSN on FPN.

### 3.4. CSN on recognition

Inspired by success of CSN on object detection, we also verify effect of CSN on other instance-related recognition tasks, such as instance segmentation and human pose estimation. Instance segmentation aims to precisely localize and predict pixel-wise mask for all instances in images; Human pose estimation aims to localize person keypoints of 17 categories accurately. He et al., [11] proposed an unified framework called Mask R-CNN to solve both tasks. We try to apply CSN to Mask R-CNN for better recognition accuracy. In details, CSN just filter out objects which are out of given scale range, and use Mask R-CNN in training and testing phases.

### 4. Experiments

#### 4.1. Common settings

We conduct experiments on three tasks, namely object detection, instance segmentation and human pose estimation respectively. Experiments are performed on COCO [17] dataset following the official dataset split. That is, models are trained on 118k train set and evaluated on the 5k validation set. Final result of detection for comparison is submitted to official test server of COCO.

**implementation details** If without specific description, following settings apply to both baseline and CSN.

All models are implemented on the same codebase for
Table 3: SOTA Comparison on COCO test-dev. w DCN indicates deformable convolution [4] is adopted in network. All methods are trained on COCO train2017. SNIP[26] and SNIPER[27] are recently proposed methods and achieve very impressive performance. CSN is integrated with FPN and surpasses previous methods hugely.

Table 4: Comparison on COCO val2017 Baseline is trained and tested with single scale (800,1333). The MS denotes multi-scale (with 7 scales for training and 9 scales for testing) and the detailed scale setting follows [11]. As is shown, CSN consistently improves APs. Methods with superscript * are evaluated on COCO test-dev.

comparison based on MXNet¹. The training and testing hyper-parameters are almost the same as Mask R-CNN [11], while some modifications were made.

Considering network structures, 2-fc layer detection head exists in all models, which is different from [12] which attaches conv5 as the hidden layers of the head. For small backbones, such as ResNet-18, ResNet-18-FPN and MobileNet-v2-FPN, the dimension of 2 fc-layers was 512 and 256; while others employed two 1024. Besides, we use Soft-NMS[2] to replace conventional NMS for post-processing after detection head. Other settings generally follows Mask R-CNN [11].

For training mechanism, all models were trained on 8 GPUs with synchronized SGD for 18 epochs. The learning rate (lr) is initialized with 0.00125 * bs which is linear with mini-batch size bs like [9]. In addition, lr will be divided by 10 at 15-th and 18-th epoch successively. Warm-up strategy [9] is also used for the 1-st epoch.

Baseline models are all trained and tested with (800,1333) when considering single-scale strategy. And multi-scale training & testing strategy follows [11]. Concretely, training scales are randomly sampled from [608,800] pixels with a step of 32, while testing scales are [400,1200] pixels with a step of 100. This results in 7 scales for training and 9 scales for testing.

The scaling factor \( \Omega \) of CSN models is \{4.0,2.0,1.0,0.5,0.25\} for both training and testing phase. CSN range is set as [16,560] by experiments(in Sec. 4.2.1).

In the following, experiments of CSN for different recognition tasks are described in details.

4.2. CSN on object detection

Detector is based on the two-stage Faster R-CNN [24] framework, i.e. the detection part of Mask R-CNN [11]. Backbones used here consists of vanilla ResNet, ResNet-FPN(ResNet with FPN) and MobileNet-v2-FPN. We will report standard COCO detection metrics, including AP(mean AP over multiple IoU thresholds), AP\(_{50}\), AP\(_{75}\), AP\(_s\), AP\(_m\), AP\(_l\)(AP at different scales). All training and testing implementation details are the same as 4.1.

¹https://mxnet.apache.org/
4.2.1 CSN range

CSN introduces consistency on scale range, and the only remaining problem is how to find the best CSN range. We propose a simple greedy policy which iteratively adjusts the upper bound and lower bound to find the one with best AP locally. Experiments on those different range candidates are performed to evaluate AP. Table 5 shows some CSN range candidates with corresponding accuracy statistics. The following describes the iterative search procedure in details.

The initial range is set to [0, 640] because of the object size limitation of COCO dataset. Firstly $s_u$ is fixed and $s_l \in \{0, 16, 32\}$ are evaluated respectively. The AP of [0, 640] is only 37.4, while AP of [16, 640] increases to 38.2 because of excluding many hard tiny objects (about 14% objects’ scales in COCO lie in [0, 16]) during training. However, further lifting $s_l$ to 32 deteriorates the AP, because too many small objects have been ignored in training phase. So the locally optimal $s_l$ is 16. Secondly $s_l$ is set to 16 and $s_u \in \{640, 560, 496, 320\}$ are evaluated severally. From [16, 640] to [16, 560], AP, AP$_m$ and AP$_l$ increase because some large objects are filtered during training. This indicates extremely large objects could disturb learning process doubtlessly. However, further reducing $s_u$ to 496 deteriorates AP, in particular AP$_m$, AP$_l$. We consider that each stage in FPN learns from objects of different sizes, e.g., the $P_n$ of original FPN receives objects which scale in [480, inf] by heuristic rule [15]. And discarding objects in [496, 560] causes $P_n$ under-fitting due to lack of training samples. Analogously, [16, 320] deteriorates AP more. So the locally optimal $s_u$ is 560. Thirdly $s_l$ was set to 560 and $s_u = 32$ is evaluated continuously. The AP hasn’t been improved. This iterative search procedure is completed up to now, and the best CSN range is [16, 560]. And CSN range will be set to [16, 560] for following experiments.

4.2.2 Main Results

The comparison of CSN with other state-of-the-art methods on COCO test-dev is shown in Table 3. To the best of our knowledge, for ResNet-50-FPN backbone (with deformable convolutions) based Faster R-CNN architecture, final mAP on COCO test-dev is 45.3%, surpassing previous detectors with same backbone and architecture by a large margin. CSN, baseline and other methods are also compared on COCO val, as shown in Table 4. With regard to ResNet-50-FPN backbone, AP 38.0 of single scale baseline which is re-implemented is comparable to the one in Detectron [8]. While multi-scale training & testing improved baseline by 2.9 point, CSN improved much more to 4.9 point. Deformable convolution (DCN) was also introduced because of the good property at modelling object deformation. With DCN integrated, original baseline goes up to 41.1, and CSN also surpasses multi-scale competitor again by 2 point gap. These prove the effectiveness and compatibility of CSN. In addition, CSN promotes a huge 14.5 point compared the SSDLite [25] version on MobileNet-v2 backbone.

4.2.3 Results on small backbones

We also evaluate CSN on other small network architectures, including vanilla ResNet-18, ResNet-18-FPN and MobileNet-v2-FPN. With respect to ResNet-18 based Faster R-CNN in Table 6, multi-scale training & testing improves the AP of single scale model by about 2 point, and CSN significantly boosts incredible 5 point additionally. The AP$_s$, AP$_m$, AP$_l$ get promotion steadily. Moreover as shown in Table 6, using FPN can improve AP of single scale model by 4 point. Multi-scale FPN further increases AP to 35.6. CSN with FPN still boosts 3.1 point. This demonstrates that CSN can bring profit to detector consistently no matter the existence of FPN. What’s more, the vanilla Faster R-CNN with CSN even surpasses the multi-scale FPN without CSN by 0.5 point, manifesting CSN’s superiority. We also apply CSN to MobileNet-v2-FPN, as shown in Table 6. CSN improves more than 2 point steadily comparing with multi-scale training & testing.

4.2.4 CSN for efficient detection

Although CSN give model impressive accuracy promotion, it may be criticized owing to its time-consuming multi-scale testing in real application. Experiments are conducted to inspect influence of CSN on tiny models. And extra comparative experiments are conducted on different single-scale testing circumstances for the same tiny model. The one is testing on the same pre-defined resolution for all images, while another is testing on original resolution for each image. All results are shown in Table 7. The trained Base-line, MS train and CSN model are the same as those in Table 6. For the 1st single-scale testing circumstance, the pre-defined resolution for Baseline and MS train models is (800, 1333), while 600 for CSN because it’s closer to the patch size 576 used in CSN training. At both testing circumstances, the AP of CSN defeats other two competitors. When changing from pre-defined resolution to original resolution, AP of Baseline and MS train declines sharply, while AP of CSN only decreases 1 point. This also proves CSN’s robustness for scale variation.

In real application, there exists strict requirement of detection accuracy for objects only in common scale range (e.g. [16, 560]). As Table 7 shows, AP$_{[16-560]}$ of CSN surpasses other two methods by a large margin. It’s because CSN training make model focus on learning object representation in this range. In summary, CSN achieves better accuracy in specific scale range while being faster with
single-scale testing. This suggests potential usage of CSN in real situation like security monitoring and etc.

### 4.3. CSN on other recognition tasks

Experiments are also performed on other two object-related recognition tasks, namely instance segmentation and human pose estimation (namely person keypoint detection). Firstly, Evaluation metrics for both are marked by AP_{mask} and AP_{kps} severally. The higher AP means more accurate for both, which is similar to detection metric. Next, for economy, Backbones used here only includes ResNet-FPN (18 and 50), and only multi-scale strategy is compared to CSN. The following details more.

#### 4.3.1 CSN on instance segmentation

**Main results**  Results of detection and instance segmentation on different backbones are shown in Table 8. As you can see, AP on mask for CSN achieves more than 1 point increment comparing with multi-scale training & testing in both shallow and deep backbones. Even if introducing DCN component, the increment is also obvious and robust. This indicates the compatibility of CSN and other strong components (e.g. DCN). It’s worth noting that CSN can also improve AR metric on objects of different sizes. This is essential to real applications.

| CSN_range | [0,640] | [16,640] | [32,640] | [16,560] | [16,496] | [16,320] | [32,560] |
|-----------|---------|---------|---------|---------|---------|---------|---------|
| AP_s      | 26.0    | 26.9    | 25.9    | 26.5    | 26.2    | 26.3    | 27.0    |
| AP_{m1}   | 39.4    | 40.1    | 40.6    | 40.5    | 39.8    | 39.1    | 40.6    |
| AP_l      | 46.6    | 47.6    | 47.9    | 47.9    | 46.7    | 46.8    | 48.5    |
| AP        | 37.4    | 38.2    | 38.1    | 38.7    | 37.9    | 37.2    | 38.4    |

Table 5: **ResNet-18-FPN trained with different CSN range**: Since object scale in COCO [17] ranges from 0 to 640, the trials of range is limited in that region. Experimental settings are the same in the table, except the CSN range differs.

#### 4.3.2 CSN on human pose estimation

Generally speaking, there are always two modes of human pose estimation: 1) From image, namely detecting person bounding-box, then cropping it from original image and resizing to standard scale, finally using backbone predicting keypoint heatmaps directly. 2) From feature map, namely detecting person instance and corresponding keypoints simultaneously, which is equivalent to predicting heatmaps from feature map at inter layer. The former mode is always more accurate than the latter, because of the explicit normalization operation on scale in image, i.e., cropping and resizing to uniform resolution. Experiments on CSN for kps are performed on COCO-kps data and Mask R-CNN structure, namely the 2nd mode. The final accuracy boosting on pose estimation in results implies the effect of implicit normalization from CSN.

**Implementation details** Original Mask R-CNN [11] models a keypoint’s location distribution on RoI as an one-hot mask, and predicts $K = 17$ heatmaps, each for one keypoint type. However the kps head (the head for human pose estimation) is too heavy for real application, especially for multi-person pose estimation. Therefore, Four modifications on head’s structure are made for both CSN and baseline model. Firstly, to better approximate person’s shape, we use RoIAlign to extract feature map of $22 \times 16$ size for each person RoI, instead of $14 \times 14$. Secondly, to re-

| MobileNet-v2-FPN | AP | AR | AP_{16,560} | AR_{16,560} |
|------------------|----|----|-------------|-------------|
| Baseline_{1.0}   | 27.6| 54.4| 31.0        | 60.3        |
| Baseline_{800}   | 32.8| 60.8| 35.9        | 65.2        |
| +MS Train_{1.0}  | 29.1| 55.7| 32.6        | 61.4        |
| +MS Train_{800}  | 32.9| 61.1| 35.9        | 65.4        |
| +CSN_{1.0}       | 32.6| 59.4| 37.1        | 66.9        |
| +CSN_{500}       | 33.6| 61.4| 37.4        | 67.1        |

Table 7: **Single scale test on MobileNet-v2-FPN**: Baseline is trained in single scale (800,1333). The multi-scale trained model and model trained with ISN are shown. Each row is tested with $\omega_0 = 1.0$ (i.e., raw image) or with short side 600 or 800, which is denoted by the subscript. Testing with raw image is much faster than with short side 800. $AP_{16,560}$ represents the AP calculated on GT boxes in [16,560].

Table 6: **Results on ResNet-18, ResNet-18-FPN and MobileNet-v2-FPN**: Baseline is trained and tested with [800,1333]. MST denotes multi-scale training & testing here. Each row adds an individual component to the 1-st row (i.e., Baseline)
Table 8: Mask Results: Results of ResNet-18,50 on COCO2017. CSN achieves more than 1 point improvement of AP$^\text{det}$ and AP$^\text{kps}$ on different backbones. Robust and comparable improvements of AR$^\text{mask}$.

Table 9: CSN v.s. MST: AP of person instance detection and keypoint detection on COCO val2017. The backbone is ResNet-18-FPN, ResNet-50-FPN without and with DCN. CSN improves more than 3.5 point on AP$^\text{kps}$ comparing with MST for both backbones. The accuracy promotion on kps is steady whether with DCN or not, proving CSN is compatible with DCN. For comparison to official Mask R-CNN, the MS Train&MS Test with modified kps head is more quick and bandwidth-saving in practice, but inferior on accuracy. The CSN model with the same modified kps head can defeat Mask R-CNN[11] easily.

Main results Results of person detection and pose estimation on different backbones are shown in Table 9. Firstly, official Mask R-CNN[11] is compared to CSN and MS train models on ResNet-50-FPN backbone in the 2-nd row in Table 9. It should be pointed out that, the modified design reduces nearly half float operation number for kps head. We observe that, the MS train model is inferior on AP$^\text{det}$, because the relative change of head capacity from reduced kps head to original detection head, training process might pay more attention to detection branch. This contrary effect is also shown in [11]. However, even with the same light-weight design, CSN still exceeds Mask R-CNN[11] by 1 point on AP$^\text{kps}$. This proves the effectiveness of CSN for human pose estimation.

Secondly, CSN boosts performance of human pose estimation on several backbones significantly, as shown in Table 9. The AP$^\text{kps}$ gets more close to kps method [31] from image, namely the aforementioned 1st mode, than Mask R-CNN[11]. This phenomenon indicates that implicit scale normalization in CN can assist kps, consistently with the explicit scale normalization in 1st mode method.

5. Conclusion

We propose a novel paradigm, consistent scale normalization to solve severe scale variation problem in instance-related vision tasks. CSN integrates image pyramid (for scale normalization) and feature pyramid (for scale normalization) and feature pyramid (for easeful learning).
inspire future work to solve other large variation problems.

References

[1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 2
[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-tnns–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 5
[3] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of the european conference on computer vision*, pages 354–370. Springer, 2016. 2
[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 5
[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 1, 2
[6] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 3
[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2
[8] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 5, 6
[9] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5
[10] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, and X. Hu. Scale-aware face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6186–6195, 2017. 2
[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988. IEEE, 2017. 1, 2, 3, 4, 5, 7, 8
[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5
[13] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the european conference on computer vision*, pages 734–750, 2018. 2
[14] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019. 4
[15] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 4, 2017. 1, 2, 6
[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1, 2
[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the european conference on computer vision*, pages 740–755. Springer, 2014. 1, 4, 7
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the european conference on computer vision*, pages 21–37. Springer, 2016. 2
[19] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 4875–4884, 2017. 2
[20] M. Najibi, B. Singh, and L. S. Davis. Autofocus: Efficient multi-scale inference. *arXiv preprint arXiv:1812.01600*, 2018. 1
[21] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4903–4911, 2017. 8
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2
[23] J. Redmon and A. Farhadi. Yolo3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2
[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 3, 5
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 354–370. Springer, 2016. 2
[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2
[27] B. Singh and L. S. Davis. An analysis of scale invariance in object detection—snip. In *Proceedings of the International Conference on Computer Vision*, pages 4510–4520, 2018. 5, 6
[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5
[29] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 2
[30] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. *arXiv preprint arXiv:1901.03278*, 2019.

[31] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision*, pages 466–481, 2018.

[32] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. *arXiv preprint arXiv:1903.00621*, 2019.