VarifocalNet: An IoU-aware Dense Object Detector

Haoyang Zhang[1], Ying Wang[2], Feras Dayoub[1], Niko Sünderhauf[1]
[1]Australian Centre for Robotic Vision, Queensland University of Technology
[2]University of Queensland
{h202.zhang, feras.dayoub, niko.suenderhauf}@qut.edu.au, ying.wang@uq.edu.au

Abstract

*Accurately ranking a huge number of candidate detections is a key to the high-performance dense object detector. While prior work uses the classification score or the combination of it and the IoU-based localization score as the ranking basis, neither of them can reliably represent the rank, and this harms the detection performance. In this paper, we propose to learn IoU-aware classification scores (**IACS**) that simultaneously represent the object presence confidence and localization accuracy, to produce a more accurate rank of detections in dense object detectors. In particular, we design a new loss function, named **Varifocal Loss**, for training a dense object detector to predict the IACS, and a new efficient star-shaped bounding box feature representation for estimating the IACS and refining coarse bounding boxes. Combining these two new components and a bounding box refinement branch, we build a new dense object detector on the FCOS architecture, what we call **VarifocalNet** or **VFNet** for short. Extensive experiments on MS COCO benchmark show that our VFNet consistently surpasses the strong baseline by ∼2.0 AP with different backbones and our best model with Res2Net-101-DCN reaches a single-model single-scale AP of 51.3 on COCO* `test-dev`*, achieving the state-of-the-art among various object detectors. Code is available at:* `https://github.com/hyz-xmaster/VarifocalNet`.

1. Introduction

Modern object detectors, regardless of being a two-stage method [1–4] or a one-stage method [5–9], usually first generate a redundant set of bounding boxes with classification scores and then deploy non-maximum suppression (NMS) to remove duplicated bounding boxes on the same object. Generally, the classification score is used to rank the bounding box in NMS, and if a bounding box at a lower rank has an intersection over union (IoU) over a certain threshold, *e.g*. 0.5, with a bounding box of a higher rank, it is removed. However, doing so, harms the detection performance. Because the classification score is not always a good estimation of the bounding box localization accuracy [10] and accurately localized detections with low classification scores may be mistakenly eliminated in NMS.

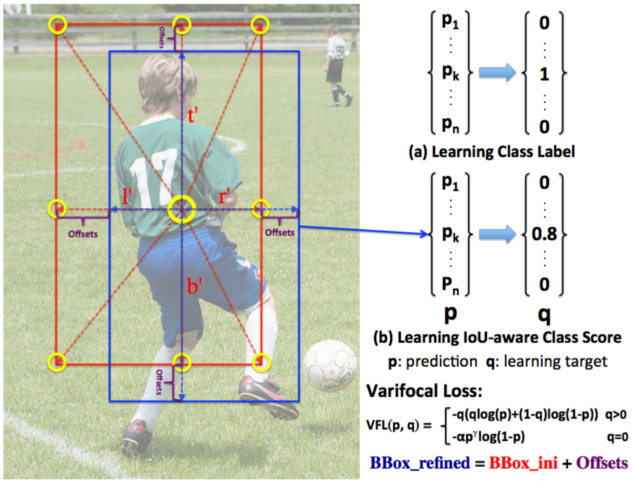
Offsets
t'
l'
r'
b'
Offsets
Offsets
Offsets
p1 ⋮ pk ⋮ pn
0 ⋮ 1 ⋮ 0
(a) Learning Class Label
p1 ⋮ pk ⋮ pn
0 ⋮ 0.8 ⋮ 0
p
q
(b) Learning IoU-aware Class Score
p: prediction q: learning target
Varifocal Loss:
VFL(p, q) = -q(qlog(p)+(1-q)log(1-p)) q>0; -αp^γlog(1-p) q=0
BBox_refined = BBox_ini + Offsets

Figure 1: An illustration of our method. Instead of learning to predict the class label (a) for a bounding box, we learn the IoU-aware classification score (**IACS**) which merges the object presence confidence and localization accuracy (b) as its detection score. We propose a varifocal loss for training the dense object detector to predict the IACS, and a star-shaped bounding box feature representation (the features at nine yellow sampling points) for estimating the IACS. With the new representation, we refine the initially regressed box (in red) into a more accurate one (in blue).

To solve the problem, existing dense object detectors predict either an additional IoU score [11] or a centerness score [9] as the localization accuracy estimation, and multiply them by the classification score to rank detections in NMS. These methods can alleviate the misalignment problem between the classification score and the object localization accuracy. However, they are sub-optimal because multiplying the two imperfect predictions together may lead to a worse rank basis and we show in Section 3 that the upper bound of the performance achieved by this kind of methods is very limited. Besides, adding an additional network branch to predict the localization score is not an elegant solution and incurs additional computation burden.

To overcome these shortcomings, we naturally would like to ask: *Instead of predicting an additional localization accuracy score, can we merge it into the classification score?* That is, predict a localization-aware or IoU-aware classification score (**IACS**) that simultaneously represents the presence of a certain object class and the localization accuracy of the generated bounding box.

In this paper, we answer the above question and make the following contributions. **(1)** We show that producing a proper score for accurately ranking a large number of candidate detections is a key to the high-performance dense object detector and the IACS is the top choice (Section 3). **(2)** We propose a new **Varifocal Loss** for training the dense object detector to regress the IACS. **(3)** We design a new efficient star-shaped bounding box feature representation for estimating the IACS and refining the bounding box. **(4)** We develop a new dense object detector based on the FCOS [9] architecture and the proposed components, named **VarifocalNet** or **VFNet** for short, to exploit the advantage of the IACS. An illustration of our method is shown in Figure 1.

In particular, the varifocal loss, inspired by the focal loss [8], is also a dynamically scaled cross entropy loss. However, it supervises the dense object detector to regress continuous values, and more distinctively it adopts an **asymmetrical** training example weighting method, as its name implicitly expresses. It down-weights only negative examples for addressing the class imbalance problem during training, and yet up-weights high-quality positive examples for generating prime detections. This focuses the training on high-quality positive examples, which is important to achieve a high detection performance.

The star-shaped bounding box feature representation is designed to efficiently encode a bounding box. It uses the features at nine fixed sampling points (yellow circles in Figure 1) to represent a bounding box with the deformable convolution [12]. Compared to the point feature used in most existing dense object detectors [7–9, 13], this feature representation can capture the geometry information of the bounding box and its nearby contextual information, which is essential for predicting an accurate IACS. It also enables us to effectively refine the initially generated coarse bounding box without losing efficiency.

To verify the effectiveness of our proposed modules, we build the VFNet based on the FCOS and evaluate it on MS COCO benchmark [14]. Experiment results show that our VFNet consistently exceeds the strong baseline by ∼2.0 AP with different backbones, and our best model with a Res2Net-101-DCN backbone reaches a single-model and single-scale AP of 51.3 on COCO `test-dev`, achieving the state-of-the-art among various object detectors.

2. Related Work

Object Detection: With the development of object detection, currently popular object detectors can be categorized by whether they use anchor boxes or not. While popular two-stage methods [3, 4] and multi-stage methods [15] usually employ anchors to generate object proposals for downstream classification and regression, anchor-based one-stage methods [6–8, 16, 17] directly classify and refine anchor boxes without object proposal generation.

More recently, anchor-free detectors have attracted substantial attention due to their novelty and simplicity. One kind of them formulates the object detection problem as a key-point or a semantic-point detection problem, including CornerNet [18], CenterNet [19], ExtremeNet [20], ObjectsAsPoints [21] and RepPoints [22]. Another type of anchor-free detectors are similar to anchor-based one-stage methods, but they remove the usage of anchor boxes. Instead, they classify each point on the feature pyramids [23] into foreground classes or background, and directly predict the distances from the foreground point to the four sides of the ground-truth bounding box, to produce the detection. The popular methods include DenseBox [24], FASF [25], FoveaBox [13], FCOS [9], ATSS [26] and SPAD [27]. We build our VFNet based on the ATSS version of FCOS due to its simplicity, high efficiency and excellent performance.

Detection Ranking Measures: In addition to the classification score, other detection ranking measures have been proposed. IoU-Net [10] uses an additional network to predict the IoU and use it to rank bounding boxes in NMS, but it still selects the classification score as the final detection score. Fitness NMS [28], IoU-aware RetinaNet [11] and [29] are similar to IoU-Net in essence, except that they multiply the predicted IoU or IoU-based ranking scores and the classification score as the ranking basis. Instead of predicting the IoU-based score, FCOS [9] predicts centerness scores to suppress the low-quality detections.

By contrast, we predict only the IACS that mixes the object presence confidence and localization accuracy as the ranking score. This avoids the use of an additional network and the worse rank basis got by multiplying the imperfect localization score and classification score together.

Encoding the Bounding Box: Extracting discriminative features to represent a bounding box is important for downstream classification and regression in object detection. In two-stage and multi-stage methods, RoI Pooling [2] or RoIAlign [4] are widely employed to extract features to describe a bounding box. But it is very time-consuming if applying them in dense object detectors. Instead, one-stage detectors generally use point features as the bounding box descriptor [7–9], due to the efficiency consideration. However, these local features fail to capture the geometry of the bounding box and essential contextual information.

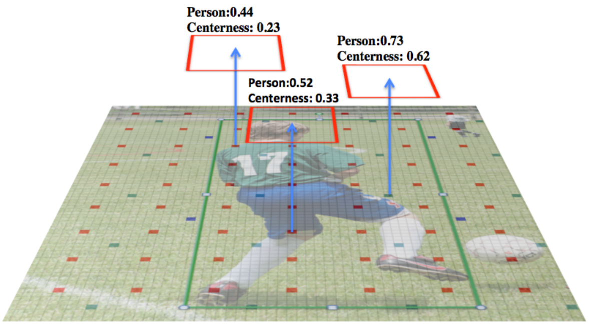

Figure 2: An example of the output from the FCOS head which includes a classification score, a bounding box and a centerness score.

Alternatively, HSD [30] and RepPoints [22] extract features at learned semantic points with the deformable convolution to encode a bounding box. However, learning to localize semantic points is challenging due to the lack of strong supervision, and the prediction of semantic points also aggravates the computation burden.

In comparison, our proposed star-shaped bounding box representation uses the features at nine sampling points to describe a bounding box. It is simple, efficient, and yet able to capture the geometry information of the bounding box and spatial context cues around it.

Generalized Focal Loss: The most similar work to ours is a concurrent work, Generalized Focal Loss (GFL) [31]. The GFL extends the focal loss [8] to a continuous version and trains the detector to predict a joint representation of localization quality and classification.

We emphasize first that our varifocal loss is a distinct function from the GFL. It weights positive and negative examples asymmetrically, whereas the GFL deals with them equally, and experiment results show that our varifocal loss performs better the GFL. Moreover, we propose an efficient star-shaped bounding box feature representation to facilitate the prediction of the IACS, and further improve the object localization accuracy through a bounding box refinement step, which are not considered in the GFL.

3. Motivation

In this section, we investigate the performance upper bound of a popular anchor-free dense object detector, FCOS [9], identify the main performance hindrance and show the importance of producing the IoU-aware classification score as the ranking basis.

We first briefly revisit the FCOS. FCOS is built on FPN [23] and its head has three branches. One predicts the classification score for each point on the feature map, one regresses the distances from the sampling image location to the four sides of a bounding box, and another one predicts the centerness score which is multiplied by the classification score to rank the bounding box in NMS. Figure 2 shows an example of the output from the FCOS head. In this paper, we actually study the ATSS version of FCOS (FCOS+ATSS) in which the Adaptive Training Sample Selection (ATSS) mechanism [26] is used to define the foreground and background points on the feature pyramids during training. We refer the reader to [26] for more details about the ATSS.

	FCOS+ATSS									
w/ctr		✓	✓	✓		✓		✓		✓
gt_ctr			✓							
gt_ctr_iou				✓						
gt_bbox					✓	✓				
gt_cls							✓	✓		
gt_cls_iou									✓	✓
AP	38.5	39.2	41.1	43.5	56.1	56.3	43.1	58.1	**74.7**	67.4

Table 1: Performance of the FCOS+ATSS on the COCO `val2017` with different oracle predictions. W/ctr means using the centerness score in inference. Please see the text for the meaning of other abbreviations.

To investigate the performance upper bound of the FCOS+ATSS (trained on COCO `train2017` [14]), we alternately replace the predicted classification score, the distance offsets and the centerness score with corresponding ground-truth values for foreground points **before** NMS, and evaluate the detection performance in terms of AP [14] on COCO `val2017`. For the classification probability vector, we implement two options, that is, replace its element at the ground-truth label position with a value of 1.0 or the IoU between the predicted bounding box and the ground-truth one (termed as gt_IoU). We also consider replacing the centerness score with the gt_IoU in addition to its true value.

The results are shown in Table 1. We can see that the original FCOS+ATSS achieves 39.2 AP. When using the ground-truth centerness score (gt_ctr) in inference, unexpectedly, only about 2.0 AP is increased. Similarly, replacing the predicted centerness score with the gt_IoU (gt_ctr_iou) only improves the AP to 43.5. This indicates that using the product of the predicted IoU score and the classification score to rank detections is certainly unable to bring significant performance improvement.

By contrast, the FCOS+ATSS with ground-truth bounding boxes (gt_bbox) achieves 56.1 AP even without centerness score (no w/ctr) in inference. But if setting the classification score as 1.0 at the ground-truth label position (gt_cls), whether or not to use the centerness score becomes important (43.1 AP vs 58.1 AP). Because the centerness score can differentiate accurate and inaccurate bounding boxes to some extent.

The most surprising result is the one obtained by replacing the classification score of the ground-truth class with the gt_IoU (gt_cls_iou), which is actually the IACS. Without the centerness score in inference, this case achieves a whopping **74.7** AP which is significantly higher that other cases. Given those results above, this in fact reveals that there already exist accurately localized bounding boxes in the large candidate pool for most ground-truth objects. The key to achieving an excellent detection performance is to accurately select those high-quality detections from the pool and these results show that the IACS is the most promising selection measure.

4. VarifocalNet

Based on the discovery above, we propose to learn the IoU-aware classification score (IACS) to rank detections. To this end, we build a new dense object detector, coined as VarifocalNet or VFNet, based on the FCOS+ATSS with the centerness branch removed. Compared with the FCOS+ATSS, it has three new components: the varifcoal loss, the star-shaped bounding box feature representation and the bounding box refinement.

4.1. Varifocal Loss

We design the *Varifocal Loss* for training the dense object detector to predict the IACS. Since it is inspired by *Focal Loss* [8], we first briefly review the focal loss.

The focal loss is proposed to address the extreme imbalance problem between foreground and background classes during the training of dense object detectors, where a vast number of easy negatives can overwhelm the cross entropy loss and dominate the gradient. It is defined as[1]:

$$\mathrm{FL}(\mathrm{p},\mathrm{y}) = \begin{cases} -\alpha(1-\mathrm{p})^{\gamma}\mathrm{log}(\mathrm{p}) & \mathrm{if} \quad \mathrm{y}=1 \\ -(1-\alpha)\mathrm{p}^{\gamma}\mathrm{log}(1-\mathrm{p}) & \mathrm{otherwise}, \end{cases} \quad (1)$$

where $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0,1]$ is the predicted probability for the foreground class. As shown in Equation 1, the focal loss adds a weighting factor (α for the foreground class and $(1-\alpha)$ for the background class) and a modulating factor ($(1-p)^{\gamma}$, p^{γ}) to the cross entropy loss. While α is used to balance the importance of positive/negative examples, the modulating factor reduces the loss contribution from easy examples and relatively increases the importance of misclassified examples. Thus, the focal loss prevents the vast number of easy negatives from overwhelming the detector during training and focuses the detector on a sparse set of hard examples.

We borrow the example weighting idea from the focal loss to address the class imbalance problem when training a dense object detector to regress the continuous IACS. However, unlike the focal loss that deals with positives and negatives equally, we treat them asymmetrically. Our varifocal loss is defined as:

$$\mathrm{VFL}(\mathrm{p},\mathrm{q}) = \begin{cases} -\mathrm{q}(\mathrm{q}\mathrm{log}(\mathrm{p}) + (1-\mathrm{q})\mathrm{log}(1-\mathrm{p})) & \mathrm{q}>0 \\ -\alpha\mathrm{p}^{\gamma}\mathrm{log}(1-\mathrm{p}) & \mathrm{q}=0, \end{cases} \quad (2)$$

where p is the predicted IACS and q is the target IoU score. For a positive training example, q is set as the IoU between the generated bounding box and the ground-truth one (gt_IoU), whereas for a negative training example, the training target q for all classes is 0. See Figure 1.

As Equation 2 shows, the varifocal loss only reduces the loss contribution from negative examples by scaling their losses with a factor of p^{γ} and does not down-weight positive examples in the same way. This is because positive examples are extremely rare compared with negative examples and we should keep their precious learning signals. On the other hand, inspired by PISA [32] and IoU-balanced loss [33], we weight the positive example with the training target q. If a positive example has a high gt_IoU, its contribution to the loss will thus be relatively big. This focuses the training on those high-quality positive examples which are more important for achieving a higher AP than those low-quality examples.

To balance the losses between positive examples and negative examples, we add an adjustable scaling factor α to the negative loss term.

4.2. Star-Shaped Box Feature Representation

We design an efficient star-shaped bounding box feature representation for estimating the IACS. It uses the features at nine fixed sampling points (yellow circles in Figure 1) to represent a bounding box with the deformable convolution. This new representation can capture the geometry information of a bounding box and its nearby contextual information, which is essential for encoding the misalignment between the predicted bounding box and the ground-truth one.

Specifically, given a sampling location (x, y) on the image plane, we first regresses an initial bounding box from it with 3x3 convolution. Following the FCOS, this bounding box is encoded by a 4D vector (l', t', r', b') which means the distance from the location (x, y) to the left, top, right and bottom side of the bounding box respectively. With this distance vector, we heuristically select nine sampling points at: (x, y), (x-l', y), (x, y-t'), (x+r', y), (x, y+b'), (x-l', y-t'), (x+l', y-t'), (x-l', y+b') and (x+r', y+b'). These nine locations are then mapped onto the feature map and features at the projecting points are convolved by the deformable convolution to represent a bounding box. Since these points are manually selected without additional prediction burden, our new representation is computation efficient.

[1] Here, the multi-classification problem is formulated as multiple binary classification sub-problems. In this paper, we follow the same practice.

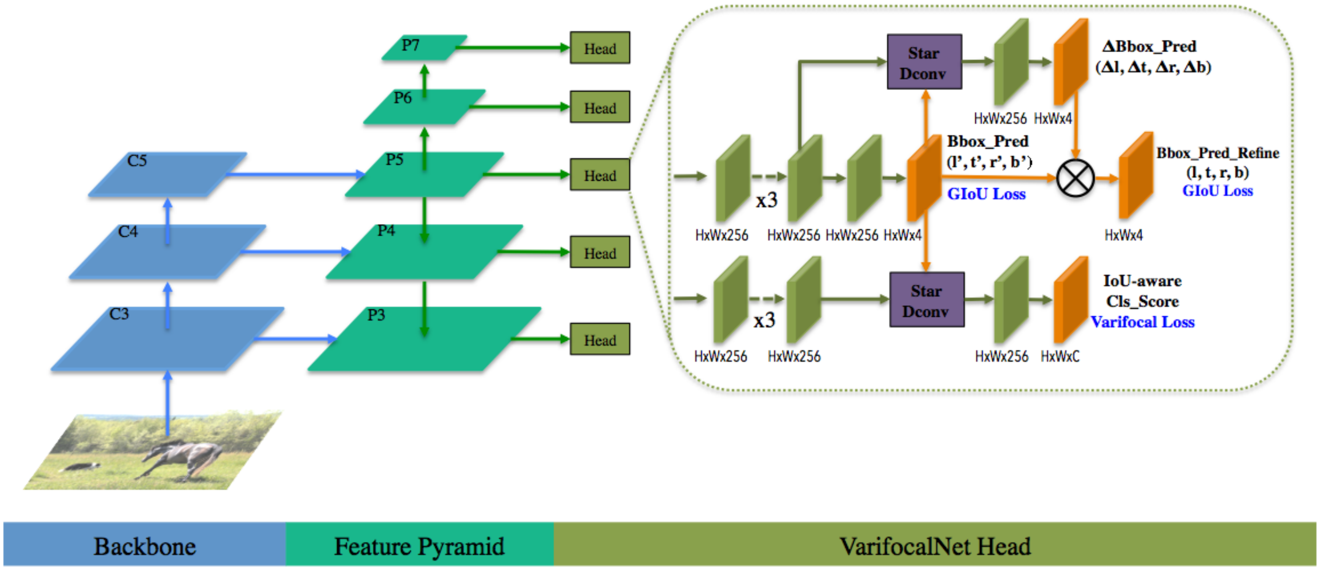

Figure 3: The network architecture of our VFNet. The VFNet is built on the FPN (P3-P7). Its head consists of two subnetworks, one for regressing the initial bounding box and refining it, and the other for predicting the IoU-aware classification score, based on a star-shaped bounding box feature representation (Star Dconv). `H×W` denotes the size of the feature map.

4.3. Bounding Box Refinement

We further improve the object localization accuracy through a bounding box refinement step. Bounding box refinement is a common technique in object detection, however, it is not widely adopted in dense object detectors due to the lack of an efficient and discriminative object descriptor. With our new efficient star representation, we can now adopt it in dense object detectors without losing efficiency.

We model the bounding box refinement as a residual learning problem. For an initially regressed bounding box (l', t', r', b'), we first extract the star-shaped representation to encode it. Then, we learn four distance scaling factors (Δl, Δt, Δr, Δb) to scale the distance vector so that the refined bounding box represented by (l, t, r, b) = (Δl×l', Δt×t', Δr×r', Δb×b') is closer to the ground truth.

4.4. VarifocalNet

Attaching the above three components to the FCOS network architecture and removing the original centerness branch, we get the *VarifocalNet*.

Figure 3 illustrates the network architecture of the VFNet. The backbone and FPN network of the VFNet are the same as the FCOS. The difference lies in the detector's head structure. The VFNet head consists of two subnetworks. The localization subnet performs bounding box regression and subsequent refinement. It takes as input the feature map from each level of the FPN and first applies **three** 3x3 conv layers with ReLU activations. This produces a feature map with 256 channels. One branch of the localization subnet convolves the feature map again and then outputs a 4D distance vector (l', t', r', b') per spatial location which represents the initial bounding box. Given the initial box and the feature map, the other branch applies a star-shaped deformable convolution [12] to the nine feature sampling points and produces the distance scaling factor (Δl, Δt, Δr, Δb) which is multiplied by the distance vector to generate the refined bounding box (l, t, r, b).

The other subnet is responsible for predicting the IACS. It has the similar architecture to the localization subnet (the refinement branch) except that it outputs a vector with C (the class number) elements per spatial location. The value of each element represents jointly the object presence confidence and localization accuracy.

Note that through the star-shaped deformable convolution, we correlates the localization network and the IACS network, which helps learn better representation for both tasks, as verified in RepPoints [22].

4.5. Loss Function and Inference

Loss Function. The training of our VFNet is supervised by the loss function:

$$\begin{aligned} \mathrm{Loss} = & \frac{1}{\mathrm{N_{pos}}} \sum_{\mathrm{i}} \mathrm{VFL}(\mathrm{p_i}, \mathrm{q_i}) \\ & + \frac{\lambda_0}{\mathrm{N_{pos}}} \sum_{\mathrm{i}} \mathrm{q_i L_{bbox}}(\mathrm{bbox'_i}, \mathrm{bbox^*_i}) \\ & + \frac{\lambda_1}{\mathrm{N_{pos}}} \sum_{\mathrm{i}} \mathrm{q_i L_{bbox}}(\mathrm{bbox_i}, \mathrm{bbox^*_i}) \end{aligned} \tag{3}$$

where p_i and q_i denote the predicted and ground-truth IACS at the location i on each level feature map of FPN, respectively. L_{bbox} is the GIoU loss [34], and $bbox'_i$, $bbox_i$ and $bbox^*_i$ represent the initial, refined and ground-truth bounding box respectively. We weight the L_{bbox} with the training target q_i, which is a value $\in (0, 1]$ for positive examples and 0 otherwise, following the practice in the FCOS. λ_0 and λ_1 are the balance weights for L_{bbox} and are empirically set as 1.5 and 2.0 respectively in this paper. N_{pos} means the number of positive examples and is used to normalize the total loss. As mentioned in Section 3, we employ the ATSS [26] to define positive and negative training examples.

Inference. The inference of the VFNet is straightforward. It involves simply forwarding an input image through the network and a NMS post-processing step for removing redundant detections.

5. Experiments

Dataset and Evaluation Metrics. We evaluate the VFNet on the challenging MS COCO 2017 benchmark [14]. Following the common practice [3, 8, 9, 26], we use the `train2017` split for training, report ablation results on the `val2017` split and compare with other detectors on the `test-dev` split by uploading the results to the evaluation server. We adopt the standard COCO-style Average Precision (AP) as the evaluation metrics.

Implementation and Training Details. We implement the VFNet with MMDetection v2.3 [35]. Unless specified, we adopt the default hyper-parameters used in MMDetection. The initial learning rate is set as 0.01 and we employ the linear warming up policy [36] to start the training where the warm-up ratio is set as 0.1. We use 8 V100 GPUs for training with a total batch size of 16 (2 images per GPU) in both ablation studies and performance comparison.

For ablation studies on the `val2017`, the ResNet-50 [37] is used as the backbone network and 1x training schedule (12 epochs) [35] is adopted. Input images are resized to a maximum scale of 1333×800, without changing the aspect ratio. Only random horizontal image flipping is used for data augmentation.

For performance comparison with the state-of-the-art on the `test-dev`, We train the VFNet with different backbone networks, including those ones with the deformable convolution layer [12, 38] (denoted as DCN) inserted. Note that when DCN is used in the backbone we also insert it into the last layers before the star deformable convolution in the VFNet Head. 2x (24 epochs) training scheme and multi-scale training (MSTrain) are adopted, where a maximum image scale for each iteration is randomly selected from a scale range. In fact, we apply two image scale ranges in experiments. For fair comparison with the baseline, we use the scale range 1333×[640:800]; out of curiosity, we also experiment with a wider scale range 1333×[480:960]. Note that even MSTrain is employed, we keep the maximum image scale to 1333×800 in inference, although a bigger scale performs slightly better (about 0.4 AP gain with 1333×900 scale).

γ	α	q weighting	AP	AP$_{50}$	AP$_{75}$
1.0	0.50	✓	41.2	59.2	44.7
1.5	0.75	✓	41.5	59.7	45.1
2.0	0.75	✓	**41.6**	59.5	45.0
2.0	0.75		41.2	59.1	44.4
2.5	1.25	✓	41.5	59.4	45.2
3.0	1.00	✓	41.3	59.0	44.7

Table 2: Peformance of the VFNet when changing the hyper-parameters (α, γ) of the varifocal loss. q weighting means weighting the loss of the positive example with the learning target q.

VFL	Star Dconv	BBox Refinement	AP	AP$_{50}$	AP$_{75}$
			39.0	57.7	41.8
✓			40.1	58.5	43.4
✓	✓		40.7	59.0	44.0
✓	✓	✓	**41.6**	59.5	45.0

Table 3: Individual contribution of the components in our method. The first row represents the results of the raw VFNet trained with the focal loss [8].

Inference Details. In inference, we forward the input image which is resized to a maximum scale of 1333×800 through the network and obtain estimated bounding boxes with corresponding IACSs. We first filter out those bounding boxes with p$_i$ ≤ 0.05 and select at most 1k top-scoring detections per FPN level. Then, the selected detections from all levels are merged and redundant detections are removed by NMS with a threshold of 0.6 to yield the final results.

5.1. Ablation Study

5.1.1 Varifocal Loss

We first investigate the effect of the hyper-parameters of the varifocal loss on the detection performance. There are two hyper-parameters: α for balancing the losses between positive examples and negative examples, and γ for down-weighting the losses of the easy negative examples. We show the performance of the VFNet in Table 2 when varying α from 0.5 to 1.5 and γ from 1.0 to 3.0 (only the results obtained with optimal α are shown). It shows that similar results above 41.0 AP are achieved and our varifocal loss is quite robust to different sets of (α, γ). Among those, α = **0.75** and γ = **2.0** work best (41.6 AP), and we adopt these two values for all the following experiments.

Method	Backbone	FPS	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Anchor-based multi-stage:								
Faster R-CNN [3]	X-101		40.3	62.7	44.0	24.4	43.7	49.8
Libra R-CNN [39]	R-101		41.1	62.1	44.7	23.4	43.7	52.5
Mask R-CNN [4]	X-101		41.4	63.4	45.2	24.5	44.9	51.8
R-FCN [40]	R-101		41.4	63.4	45.2	24.5	44.9	51.8
G-RMI [41]	Ensemble		41.6	61.9	45.4	23.9	43.5	54.9
TridentNet [42]	R-101		42.7	63.6	46.5	23.9	46.6	56.6
Cascade R-CNN [15]	R-101		42.8	62.1	46.3	23.7	45.5	55.2
SNIP [43]	R-101		43.4	65.5	48.4	27.2	46.5	54.9
Anchor-based one-stage:								
SSD512 [7]	R-101		31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [6]	DarkNet-53		33.0	57.9	34.4	18.3	35.4	41.9
DSSD513 [44]	R-101		33.2	53.3	35.2	13.0	35.4	51.1
RefineDet [45]	R-101		36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [8]	R-101		39.1	59.1	42.3	21.8	42.7	50.2
FreeAnchor [16]	R-101		43.1	62.2	46.4	24.5	46.1	54.8
GFL [31]	R-50		43.1	62.0	46.8	26.0	46.7	52.3
GFL [31]	R-101		45.0	63.7	48.9	27.2	48.8	54.5
GFL [31]	X-101-32x4d		46.0	65.1	50.1	28.2	49.6	56.0
GFL [31]	R-101-DCN		47.3	66.3	51.4	28.0	51.1	59.2
GFL [31]	X-101-32x4d-DCN		48.2	67.4	52.6	29.2	51.7	60.2
Anchor-free key-point:								
ExtremeNet [20]	Hourglass-104		40.2	55.5	43.2	20.4	43.2	53.1
CornerNet [18]	Hourglass-104		40.5	56.5	43.1	19.4	42.7	53.9
Grid R-CNN [46]	X-101		43.2	63.0	46.6	25.1	46.5	55.2
CenterNet [18]	Hourglass-104		44.9	62.4	48.1	25.6	47.4	57.4
RepPoints [22]	R-101-DCN		45.0	66.1	49.0	26.6	48.6	57.5
Anchor-free one-stage:								
FoveaBox [13]	X-101		42.1	61.9	45.2	24.9	46.8	55.6
FSAF [25]	X-101-64x4d		42.9	63.8	46.3	26.6	46.2	52.7
FCOS [9]	R-101		43.0	61.7	46.3	26.0	46.8	55.0
SAPD [27]	R-101		43.5	63.6	46.5	24.9	46.8	54.6
SAPD [27]	R-101-DCN		46.0	65.9	49.6	26.3	49.2	59.6
Baseline:								
ATSS [26]	R-101	17.5	43.6	62.1	47.4	26.1	47.0	53.6
ATSS [26]	X-101-64x4d	8.9	45.6	64.6	49.7	28.5	48.9	55.6
ATSS [26]	R-101-DCN	13.7	46.3	64.7	50.4	27.7	49.8	58.4
ATSS [26]	X-101-64x4d-DCN	6.9	47.7	66.5	51.9	29.7	50.8	59.4
Ours:								
VFNet	R-50	19.3	44.3/44.8	62.5/63.1	48.1/48.7	26.7/27.2	47.3/48.1	54.3/54.8
VFNet	R-101	15.6	46.0/46.7	64.2/64.9	50.0/50.8	27.5/28.4	49.4/50.2	56.9/57.6
VFNet	X-101-32x4d	13.1	46.7/47.6	65.2/66.1	50.8/51.8	28.3/29.4	50.1/50.9	57.3/58.4
VFNet	X-101-64x4d	9.2	47.4/48.5	65.8/67.0	51.5/52.6	29.5/30.1	50.7/51.7	58.1/59.7
VFNet	R2-101 [47]	13.0	48.4/49.3	66.9/67.6	52.6/53.5	30.3/30.5	52.0/53.1	59.2/60.5
VFNet	R-50-DCN	16.3	47.3/48.0	65.6/66.4	51.4/52.3	28.4/29.0	50.3/51.2	59.4/60.4
VFNet	R-101-DCN	12.6	48.4/49.2	66.7/67.5	52.6/53.7	28.9/29.7	51.7/52.6	61.0/62.4
VFNet	X-101-32x4d-DCN	10.1	49.2/50.0	67.8/68.5	53.6/54.4	30.0/30.4	52.6/53.2	62.1/62.9
VFNet	X-101-64x4d-DCN	6.7	49.9/50.8	68.5/69.3	54.3/55.3	30.7/31.6	53.1/54.2	62.8/64.4
VFNet	R2-101-DCN [47]	10.3	50.4/**51.3**	68.9/**69.7**	54.7/**55.8**	31.2/**31.9**	53.7/**54.7**	63.3/**64.4**

Table 4: Performance (single-model single-scale) comparison with state-of-the-art detectors on MS COCO `test-dev`. VFNet consistently outperforms the strong baseline ATSS by ∼2.0 AP. Our best model reaches 51.3 AP, achieving the stat-of-the-art among various object detectors. 'R': ResNet. 'X': ResNeXt. R2: Res2Net. 'DCN': Deformable convolution network. '/' separates results of the MSTrain image scale range 1333×[640:800] / 1333×[480:960].

Method	AP	AP_{50}	AP_{75}
RetinaNet [8] + FL	36.5	55.5	38.8
RetinaNet [8] + GFL	37.3	56.4	40.0
RetinaNet [8] + VFL	**37.4**	56.5	40.2
FoveaBox [13] + FL	36.3	56.3	38.3
FoveaBox [13] + GFL	36.9	56.0	39.7
FoveaBox [13] + VFL	**37.2**	56.2	39.8
RepPoints [22] + FL	38.3	59.2	41.1
RepPoints [22] + GFL	39.2	59.8	42.5
RepPoints [22] + VFL	**39.7**	59.8	43.1
VFNet + FL	40.0	58.0	43.2
VFNet + GFL	41.1	58.9	42.2
VFNet + VFL	**41.6**	59.5	45.0

Table 5: Comparison of performances when applying the focal loss (FL) [8], the generalized focal loss (GFL) [31] and our varifocal loss (VFL) to existing popular dense object detectors and our VFNet.

We also investigate the effect of weighting the loss of the positive example with the training target q, termed as *q weighting*. The fourth row in Table 2 shows the performance of the optimal set of (α, γ) without q weighting and 0.4 AP drop is observed (41.2 AP v.s 41.6 AP). This confirms the positive effect of q weighting.

5.1.2 Individual Component Contribution

We study the impact of the individual component of our method and results are shown in Table 3. The first row shows the performance of our raw VFNet trained with the focal loss and 39.0 AP is acquired. Replacing the focal loss with our varifocal loss, the performance is improved to 40.1 AP. By adding the star-shaped representation and bounding box refinement modules, the performance is further boosted to 40.7 AP and 41.6 AP respectively. These results verify the effectiveness of the three modules in our VFNet.

5.2. Comparison with State-of-the-Art

We compare our VFNet with recent state-of-the-art detectors on the COCO `test-dev` split. We select the ATSS [26] as our baseline because it is also built on the FCOS architecture, except that it regresses the bounding box from a anchor box instead of a point.

Table 4 presents the results. Compared with the strong baseline ATSS, our VFNet achieves $\sim$2.0 AP gaps with different backbones, *e.g*. 46.0 AP vs. 43.6 AP with the ResNet-101 backbone. This validates the contribution of our method. Compared to the concurrent work, the GFL [31] (whose MSTrain scale range is 1333x[480:800]), our VFNet is consistently better than it by a considerable margin. Meanwhile, our best model trained with Res2Net-101-DCN [47] achieves a single-model single-scale AP of 51.3, surpassing all recent popular object detectors. Qualitative detection examples of applying this model on the COCO `test-dev` can be found in Figure 4.

We also report the inference speed of the VFNet in terms of frame per second (FPS). Since it is difficult to get the speed of all the listed detectors under exactly same settings, we only compare our method with the baseline ATSS. The inference speed is tested on a Nvidia V100 GPU. It can be seen that our VFNet is very efficient, *e.g*. achieving 44.8 AP at 19.3 FPS, and only incurs small additional computation overhead compared to the baseline.

Finally, comparing the results of our VFNet trained with different scale ranges, we find that the performance gaps are unexpected and interestingly they roughly grow along the capacity of the backbone networks. This suggests that we should increase the training image scale range to fit higher capacity backbones in order to achieve better detection performance, which is consistent with the insights in [48].

5.3. Generality and Superiority of Varifocal Loss

To verify the generality of our varifocal loss, we apply it to some existing popular dense object detectors, including RetinaNet [8], FoveaBox [13] and RepPoints [22], and evaluate the performance on the `val2017` split. We simply replace the focal loss (FL) [8] used in these dense object detectors (ResNet-50 backbone) with our varifocal loss for training without any other changes. To further compare with the concurrent work, the GFL [31], we also train these detectors with the generalized focal loss (GFL).

Table 5 shows the results (4 images/GPU x 4 GPUs training is used here). We can see that our varifocal loss can improve RetinaNet and FoveaBox by 0.8 AP and 0.9 AP respectively. When it comes to RepPoints, the gain increases to 1.4 AP. This shows that our varifocal loss can easily bring considerable performance boost to existing dense object detectors by simply applying it in the training. Compared to the GFL, our varifocal loss performs better than it in all cases, evidencing the superiority of our varifocal loss.

Additionally, we train our VFNet with the FL and GFL for further comparison. Results are shown in the last section of Table 5 and the consistent advantage of our varifocal loss over the FL and GFL can be observed.

6. Conclusion

In this paper, we propose to learn the IACS for ranking detections. We first verify the importance of producing the IACS to rank bounding boxes and then develop a dense object detector, VarifocalNet, to exploit the advantage of the IACS. In particular, we design a varifocal loss for training the detector to predict the IACS, and an efficient star-shaped bounding box feature representation for estimating the IACS and refining the coarse bounding box. Experiments on the MS COCO benchmark prove the effectiveness of our proposed modules and show that our VarifocalNet achieves the stat-of-the-art among various object detectors.

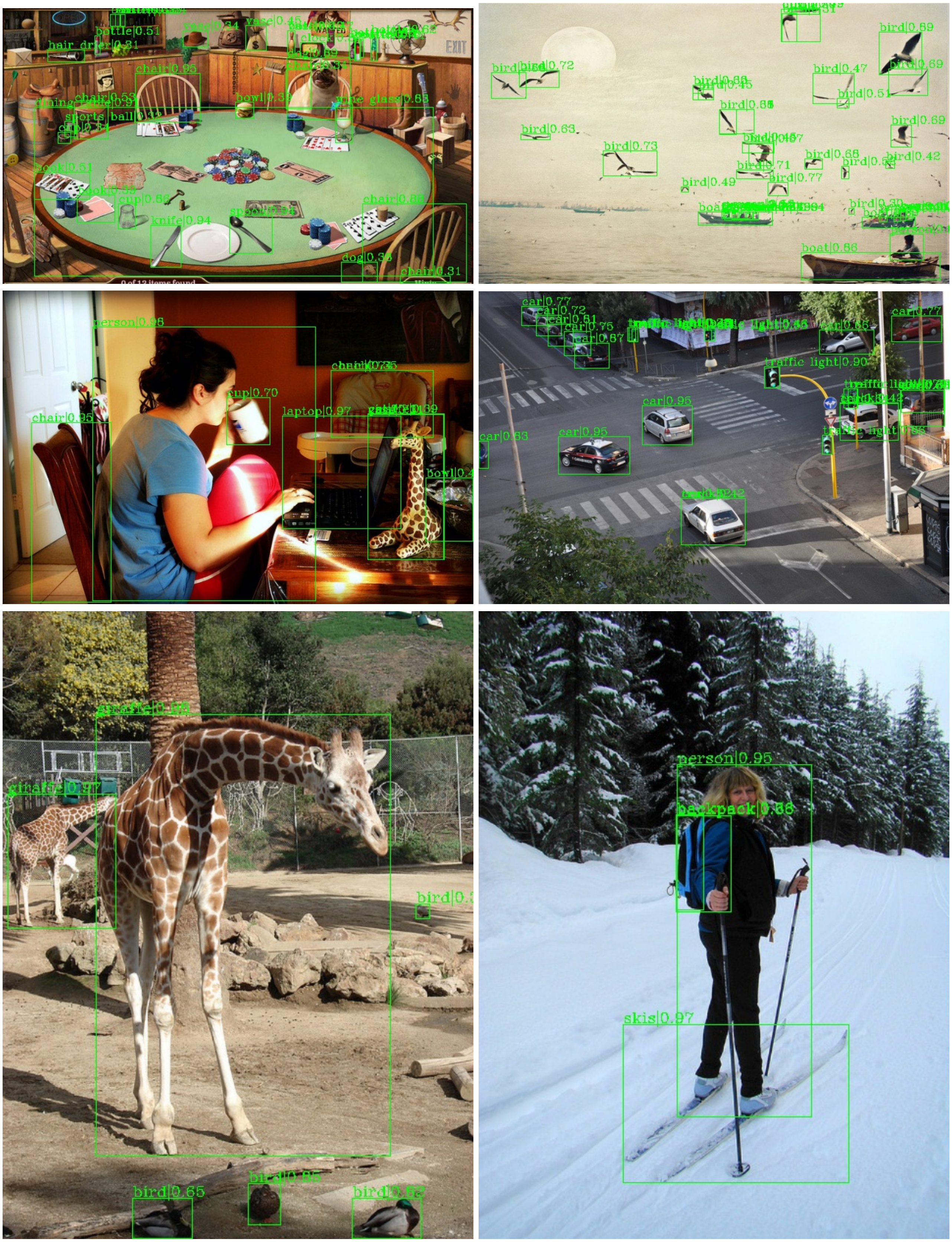

Figure 4: Detection examples of applying our best model on COCO `test-dev`. The score threshold for visualization is 0.3.

References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 1

[2] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 2

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 6, 7

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 7

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[6] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2, 7

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*. Springer. 1, 2, 7

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 2, 3, 4, 6, 7, 8

[9] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2, 3, 6, 7

[10] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018. 1, 2

[11] Shengkai Wu, Xiaoping Li, and Xinggang Wang. Iou-aware single-stage object detector for accurate localization. *Image and Vision Computing*, 2020. 1, 2

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 5, 6

[13] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 2020. 2, 7, 8

[14] 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*, 2014. 2, 3, 6

[15] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 2, 7

[16] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *NIPS*, 2019. 2, 7

[17] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *CVPR*, 2019. 2

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 2, 7

[19] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 2

[20] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2, 7

[21] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[22] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, 2019. 2, 3, 5, 7, 8

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 3

[24] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015. 2

[25] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019. 2, 7

[26] 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*, 2020. 2, 3, 6, 7, 8

[27] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *ECCV*, 2020. 2, 7

[28] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, 2018. 2

[29] Zhiyu Tan, Xuecheng Nie, Qi Qian, Nan Li, and Hao Li. Learning to rank proposals for object detection. In *ICCV*, 2019. 2

[30] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *ICCV*, 2019. 3

[31] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *arXiv preprint arXiv:2006.04388*, 2020. 3, 7, 8

[32] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. In *CVPR*, 2020. 4

[33] Shengkai Wu and Xiaoping Li. Iou-balanced loss functions for single-stage object detection. *arXiv preprint arXiv:1908.05641*, 2019. 4

[34] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 6

[35] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[36] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[38] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 6

[39] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019. 7

[40] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 7

[41] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017. 7

[42] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019. 7

[43] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018. 7

[44] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. In *CoRR*, 2017. 7

[45] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018. 7

[46] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *CVPR*, 2019. 7

[47] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 7, 8

[48] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020. 8