Robustizing Object Detection Networks Using Augmented Feature Pooling

Takashi Shibata[1], Masayuki Tanaka[2], and Masatoshi Okutomi[2]

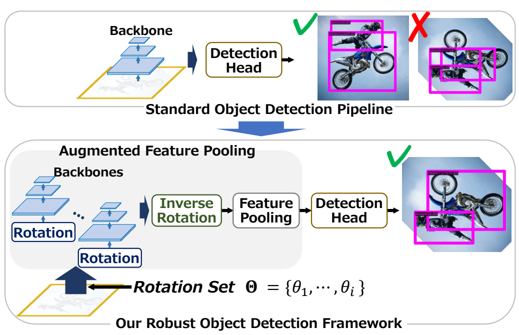
[1] NTT Corporation, Kanagawa, Japan, t.shibata@ieee.org
[2] Tokyo Institute of Technology, Tokyo, Japan

Abstract. This paper presents a framework to robustize object detection networks against large geometric transformation. Deep neural networks rapidly and dramatically have improved object detection performance. Nevertheless, modern detection algorithms are still sensitive to large geometric transformation. Aiming at improving the robustness of the modern detection algorithms against the large geometric transformation, we propose a new feature extraction called augmented feature pooling. The key is to integrate the augmented feature maps obtained from the transformed images before feeding it to the detection head without changing the original network architecture. In this paper, we focus on rotation as a simple-yet-influential case of geometric transformation, while our framework is applicable to any geometric transformations. It is noteworthy that, with only adding a few lines of code from the original implementation of the modern object detection algorithms and applying simple fine-tuning, we can improve the rotation robustness of these original detection algorithms while inheriting modern network architectures' strengths. Our framework overwhelmingly outperforms typical geometric data augmentation and its variants used to improve robustness against appearance changes due to rotation. We construct a dataset based on MS COCO to evaluate the robustness of the rotation, called COCO-Rot. Extensive experiments on three datasets, including our COCO-Rot, demonstrate that our method can improve the rotation robustness of state-of-the-art algorithms.

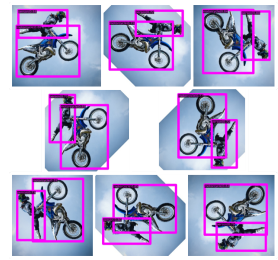
1 Introduction

There has been remarkable progress in object detection by modern network architectures [3,32,45], large image datasets with accurate annotations [16,17,33], and sophisticated open-sources [1,5,41,54]. Despite these successes, a significant issue still remains; it is sensitive to unexpected appearance changes in the wild, such as geometric transformation, occlusions, and image degradation. Particularly, rotation robustness is simple yet significant for object detection. In such cases as first-person vision, drone-mounted cameras, and robots in accidents and disasters, images are taken with unexpected camera poses and often contain large rotations.

A typical approach to improve robustness to geometric transformation is data augmentation (DA) [10,11,30,46] and test-time augmentation (TTA [21,47]).



(a) An overview of our framework



(b) Example results by our framework with Deformable-DETR [65]

Fig. 1. An overview of our proposed framework (a) and example results by our framework (b). The rotation robustness of a modern object detection network can be substantially improved by our framework based on augmented feature pooling.

Although the DA and TTA are essential learning protocols for image classification, they are powerless for rotation transformation in object detection. The reason for this is that a bounding box for an augmented image with rotation becomes much looser than the originally annotated bounding box, as we will describe in details in Sec. 3. The loosened bounding box includes a large area of background, which dramatically harm training and inference performance. This loosened bounding box problem is a common and significant challenge in object detection with the large geometric transformations such as rotation. As we will show later, DA and TTA cannot overcome the loosened bounding box problem.

A further challenge in improving the robustness of the object detection network to geometric transformation is the orientation bias of backbone feature extraction. The weight of the common backbones for object detection networks, e.g. ResNet [21] and Swin Transformer [37], are optimized for the frontal direction due to the orientation bias of the training data. Those standard backbones cannot be directly applicable to object detection tasks with arbitrary rotations. There is a strong demand for a general framework that can easily inherit the strengths of highly expressive backbones and modern object detection architectures while improving robustness to geometric transformations.

We propose a robust feature extraction framework for large geometric transformation based on augmented feature pooling. This paper focuses on rotation as a simple-yet-influential case of geometric transformation, while our framework is applicable to any kind of geometric transformations. The key is to integrate feature maps obtained from geometrically augmented images before feeding it to a detection head. It can be achieved by adding two processes: inverse rotation and feature pooling as shown in Fig. 1 (a). Examples of bounding boxes detected by our framework with Deformable DETR [65] are shown in Fig. 1 (b). We aim to improve robustness without additional annotation cost, so we only use the already annotated bounding boxes. Despite that, our proposed augmented feature pooling can substantially improve the robustness against rotation by simply adding a few lines of code to the original implementation of the existing method, and fine-tuning the model while freezing the backbone parameters. It can be easily applied to highly expressive backbones with many parameters, e.g. Swin-Transformer [37], because the proposed method does not require backbone optimization.

Our main contributions are summarized as follows[3]: 1) We propose a rotation robust feature extraction framework based on augmented feature pooling, which is applicable to various modern object detection networks. 2) We conducted preliminary experiments for investigating the problems of object detection networks on rotation robustness. 3) We constructed an object detection dataset with arbitrary rotations using MS COCO [33] for evaluating the robustness against the rotation. 4) Extensive experiments on three datasets demonstrate that our method can significantly improve the performance of state-of-the-art methods.

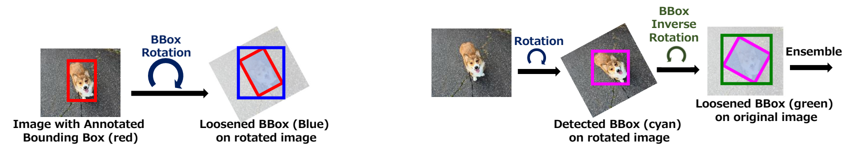
2 Related Works

Object Detection. The architectures of recent object detection consist of three components: backbone, neck, and detection head. Based on the detection head's architecture, the existing object detection algorithms can be classified into single-stage detectors [6,19,32,35,42–44] and two-stage detectors [2,4,22,39,40,45,49, 53]. While anchors are widely used, anchor-free approaches [29,48,59,64] and keypoint-based approaches [28,62] have been proposed. Beyond those CNN-based methods, Transformers have also been employed in detection networks, combining a transformer-based architecture [3,13,65] with a CNN-based backbone [12,18,21,36,56] or using a transformer-based backbone [37]. Those methods implicitly assume that the target objects are facing in the front.

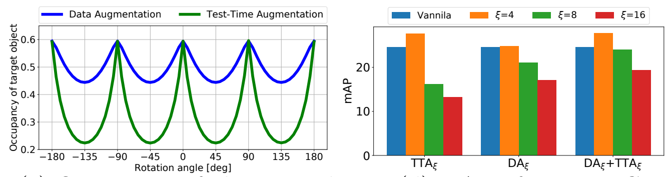
Data Augmentation and Test Time Augmentation. Data augmentation (DA) has become essential for training protocol. Learnable data augmentation algorithms by reinforcement learning and random search have been proposed [10,11,30]. Data augmentation is also effective at inference, which is called Test-Time Augmentation (TTA) [21,47]. If those DA and TTA for the classification task are naively applied to the rotated-object detection task, the detection performance will be significantly degraded because the augmentation makes the bounding box loose. Recently, a DA algorithm [25] to handle that problem has been proposed to approximate the bounding box with an inscribed ellipse to improve the robustness for small rotation. In contrast, our proposed method can be more robust to larger rotations.

Rotation-Invariant CNNs and Datasets. Rotation invariance is a fundamental challenge in pattern recognition, and many approaches have been proposed. Aiming at extracting features invariant to affine transformations including rotation, various network architectures have been proposed [7–9,26,38,50,52, 57,60,61,63]. Alignment-based approaches [23,24,51] have also been presented. Those methods are not directly applicable to the state-of-the-art object detection algorithms because those methods do not support the latest advantages including transformer-based approaches [37,65].

[3] Our code of will be available at http://www.ok.sc.e.titech.ac.jp/res/DL/index.html



(a) Data augmentation in training (b) Test-time augmentation in inference



Num. of Aug.	Rotation set Θ [deg]
$\xi = 4$	-90, 0, 90, 180
$\xi = 8$	-135, -90, $\cdots$, 180
$\xi = 16$	-157.5, -135, $\cdots$, 180

(c) Occupancy of object mask (d) mAP of Faster-RCNN [45] with DA and TTA

Fig. 2. Loosened bounding-box problem of DA and TTA.

For specific applications such as remote sensing focusing on rotation robustness, network architectures and image datasets have been designed based on an oriented bounding box [14, 15, 20, 34, 55, 58], where its rotation angle information is annotated in addition to the center position, width, and height. While the oriented bounding box is practical for these specific applications, it is not applicable to standard object detection datasets [16, 17, 33]. It is also difficult to share the advantages of modern object detection developed for the standard datasets.

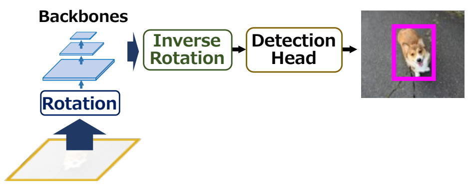
3 Two Challenges of Object Detection for Rotation Robustness

We discuss two challenges of object detection for rotation: the loosened bounding boxes and the sensitivity of backbone feature extraction on object detection task.

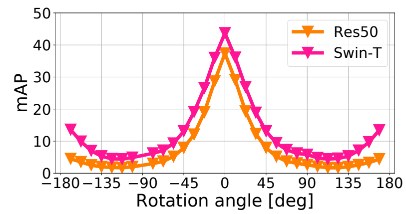
Loosened Bounding Box. When we apply geometrical data augmentation, we need to generate a new bounding box for the rotated image. If there is no segmented mask along object boundary, we have to generate a new bounding box from the originally annotated bounding box. The bounding rectangle of the rotated bounding box is commonly used as the new bounding box.

Consequently, geometrical transformation of the data augmentation (DA) and the test time augmentation (TTA) generates loosened bounding boxes, as shown in Fig. 2 (a) and (b). To evaluate the looseness of the generated bounding box, we measured the occupancy of the target object in the generated bounding box for each rotation angle during training with DA and inference with TTA[4] using MS COCO [33] as shown in Fig. 2 (c). These analysis show that (i) the loosened bounding box problem can only be avoided at integer multiples of 90

[4] Note that the TTA curve assumes that each inference before ensemble is ideal, and thus this occupancy is the upper bound.



(a) Standard object detection pipeline with rotation and inverse rotation



(b) Detection accuracy (mAP) of (a) with respect to the rotation angle.

Fig. 3. Rotation sensitivity for backbone feature extraction.

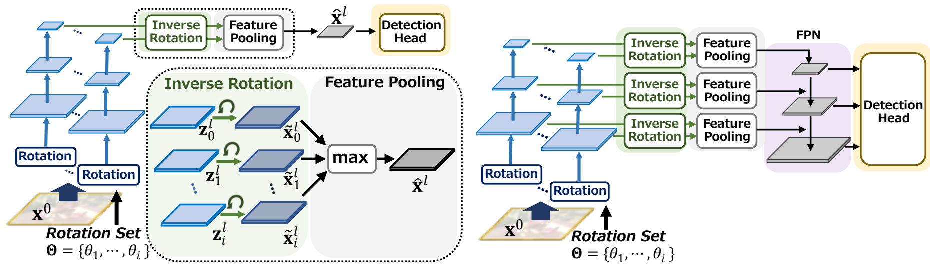
degrees, where the occupancy does not decrease, (ii) the looseness is dramatically increased where there is a deviation from those four angles.

To demonstrate the harm of the loosened bounding box for DA and TTA, we measured the mAP of Faster-RCNN [45] with DA and TTA on the rotated version of MS COCO val dataset (we will describe details in Sec. 5.1) as shown in Fig. 2 (d). For DA and TTA, we evaluated mAP using three sets of rotation angles for data augmentations denoted by ξ=4, 8, and 16, where ξ is the number of augmentation. Here, angles are assigned at equal intervals from all directions, depending on ξ. More specifically, those three rotation sets are given by Θ=$[-180, -90, 0, 90]$ for ξ=4, Θ=$[-180, -135, \cdots, 135]$ for ξ=8, and Θ=$[-180, -157.5, \cdots, 157.5]$ for ξ=16, respectively. As shown in in Fig. 2 (d), DA, TTA and those combination are only effective for $\xi = 4$ because the loosened bounding box degrades the training and inference performances for any angle except for the integer multiples of 90 degrees. The performance of DA and TTA (except for $\xi = 4$) become worse than naive fine tuning where the detection head is naively refined using training data containing arbitrary rotation.

Sensitivity of Backbone Feature Extraction. Backbone feature extraction of the object detection network is also sensitive to rotation transformations, i.e. the backbone feature extraction is not rotation invariant. When the rotated image is used as the input image, as shown in Fig 3 (a), the feature map obtained by the backbone feature map is also rotated. Even though the rotated feature map is aligned by inverse rotation, the feature map still deviates significantly from the feature map extracted from the original input image. As a result, the rotation of the input image dramatically reduces the detection accuracy (see Fig 3 (b)). Surprisingly, the sensitivity of the backbone to rotation is a common challenge, not only in commonly used backbones like the ResNet50 [21], but also in modern transformers like the Swin-Transformer [37].

4 Proposed Rotation Robust Object Detection Framework

Our proposed method aims to improve the robustness to large geometric transformations such as rotation while inheriting the strengths and weights of existing



(a) Our framework with typical object detection network

(b) Proposed framework with feature pyramid network (FPN)

Fig. 4. Proposed augmented feature pooling architecture. Feature maps obtained from the rotated images are inversely rotated and integrated by feature pooling. Then, the integrated features are fed into a detection head or FPN.

detection networks, avoiding the loosened bounding box problem and the sensitivity of backbone feature extraction. The key is to introduce augmented feature pooling, which integrates the set of the feature maps obtained from the rotated images before feeding it to the detection head. In the following, we describe our augmented feature pooling and its extension to Feature Pyramid Network [31]. Then, we explain how to extend our framework to transformer-based backbones such as Swin Transformer [37], and discuss its application to modern object detection networks.

4.1 Architecture of Augmented Feature Pooling

Figure 4 shows an overview of our proposed augmented feature pooling. Our augmented feature pooling is a simple architecture that inserts the inverse rotation and the feature pooing between the backbone and the detection head. Let $\mathbf{x}^0$, $\mathbf{x}^l$, and F^l be an input image, the l-th stage's feature map, and the backbone of l-th stage, respectively. The l-th stage's feature map $\mathbf{x}^l$ is obtained from (l–1)-th stage's feature map $\mathbf{x}^{l-1}$ and l-th stage's backbones F^l as follows:

$$\mathbf{x}^l = F^l \circ \mathbf{x}^{l-1} = F^l(\mathbf{x}^{l-1}), \tag{1}$$

where "$\circ$" is composing operator.

We start our discussion with our proposed augmented feature pooling with a standard detection architecture shown in Fig. 4 (a). We generate a set of augmented feature maps by the rotation angle θ defined by the rotation set $\Theta = \{\theta_1, \cdots, \theta_i\}$. To obtain this set of the augmented feature maps, we first generate the set of rotated images $Z^0 = \{\mathbf{z}_0^0, \cdots \mathbf{z}_i^0\}$, where the i-th rotated image $\mathbf{z}_i^0$ is generated by $\mathbf{z}_i^0 = R_{\theta_i}(\mathbf{x}^0)$ using the rotation operator R_θ. The rotation operator R_θ represents the rotation within the image plane by angle theta around the image center. Each of these rotated images $\mathbf{z}_i^0$ is fed to the backbone $F = F^l \circ F^{l-1} \circ \cdots \circ F^1$, resulting in the set of rotated feature maps $Z^l = \{\mathbf{z}_0^l, \cdots \mathbf{z}_i^l\}$.

Inverse Rotation. Alignment of feature maps is essential for feature pooling from the augmented feature maps because the object detection task simultaneously estimates the bounding box location with the class label. A set of the aligned feature maps $\tilde{X}^l$ corresponding to each augmentation is obtained by the inverse rotation $R_{-\theta_i}(\cdot)$ as follows:

$$\tilde{X}^l = \{\tilde{\mathbf{x}}_0^l, \cdots, \tilde{\mathbf{x}}_i^l\} = \{R_{-\theta_0}(\tilde{\mathbf{z}}_0^l), \cdots, R_{-\theta_i}(\tilde{\mathbf{z}}_i^l)\}. \tag{2}$$

Feature Pooling. Our proposed feature pooling performs a element-wise max pooling from the set of aligned feature maps as

$$\left(\hat{\mathbf{x}}^l\right)_k = \max_i \left(\tilde{\mathbf{x}}_i^l\right)_k, \tag{3}$$

where k is an index of an element of the feature map, and $(\hat{\mathbf{x}}^l)_k$ and $(\tilde{\mathbf{x}}_i^l)_k$ are the k-th element of $\hat{\mathbf{x}}^l$ and $\tilde{\mathbf{x}}^l$, respectively. From Eqs. (1), (2) and (3), our augmented feature pooling with the rotation set Θ is formally given by

$$\left(\hat{\mathbf{x}}^l\right)_k = \max_{\theta \in \Theta} \left(R_{-\theta}(F^l \circ \cdots \circ F^1 \circ R_\theta(\mathbf{x}))\right)_k. \tag{4}$$

In a typical object detection task, the extracted raw feature map $\mathbf{x}^l$ is used as an input for the detection head. Our proposed method feeds the pooled feature map $\hat{\mathbf{x}}^l$ to the detection head instead of the raw feature map $\mathbf{x}^l$ [5].

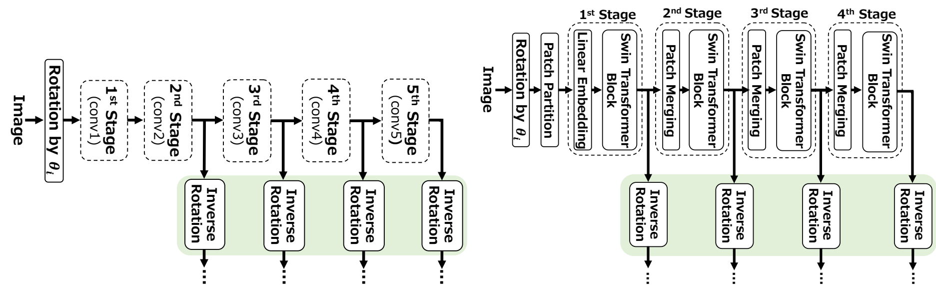
Extension to Feature Pyramid Network. Our proposed framework can be easily extended to Feature Pyramid Networks (FPN) [31] as shown in Fig. 4 (b). The inverse rotation and the feature pooling are applied to the augmented feature maps for each stage, and those pooled feature maps are fed to the FPN module. The set of pooled feature maps is denoted as $\{\hat{\mathbf{x}}^0 \cdots \hat{\mathbf{x}}^m \cdots \hat{\mathbf{x}}^l\}$. Using Eq. (4), the m-th stage's pooled feature map $\hat{\mathbf{x}}^m$ is formally represented as follows:

$$\left(\hat{\mathbf{x}}^m\right)_k = \max_{\theta \in \Theta} \left(R_{-\theta}(F^m \circ \cdots \circ F^1 \circ R_\theta(\mathbf{x}))\right)_k. \tag{5}$$

Rotation Set Designs. By designing the rotation set Θ, our proposed method can control whether to focus on a specific angle range or robust to arbitrary rotation. For example, if the rotation set Θ is uniform and dense, the robustness against arbitrary angles is improved, which is the main focus of this paper. On the other hand, if the rotation set Θ is intensively sampled around a target angle, e.g. 0 [deg] for the case where the target object is approximately facing in the front, the robustness of object detection accuracy around the target angle is improved. We will discuss the effectiveness of the rotation set designs in Sec. 5.2.

Beyond CNNs: Transformer-Based Backbone. Our proposed method can also be applied to transformer-based backbones with spatial structures such as Swin Transformer [37]. Figure 5 shows the details of the CNN-based and the Swin Transformer-based architectures. When FPN is used together with CNN-based backbones, e.g. ResNet [21], ResNeXt [56], our augmented feature pooling

[5] The dimensions of feature map $\mathbf{x}^l$ are the same as the original backbones.



(a) Our framework with CNN-based backbone

(b) Our framework with Swin Transformer-based backbone

Fig. 5. Extension to transformer-based backbone.

is applied to the feature maps obtained from Stages-2, 3, 4, and 5. On the other hand, for Swin Transformer, our augmented feature pooling is applied to the feature map immediately after the *Swin-Transformer Block* of each stage, i.e. just before the *Patch Merging*. The proposed method can be easily applied to the transformer-based backbone with spatial structure and thus inherits their rich feature representation and pre-trained weight.

4.2 Applying to Object Detection Networks

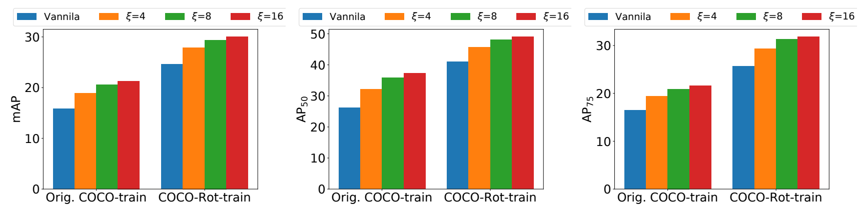
Our proposed framework is applicable to various types of detection heads such as single-stage detectors [32,42], two-stage detectors [45], and transformer-based detectors [65] without any changes in those detection head's architectures. Our framework aims to improve the robustness of rotation-sensitive detectors while taking advantage of the weight of the pre-trained backbones. We consider this robustness improvement as a downstream task, freezing the backbone and optimizing only the parameters of the detection head. Limiting the optimization parameters to the detection head allows us to quickly achieve robustness against the rotation transformation with much less computation than optimizing all parameters including the backbone.

5 Experiments

5.1 Setting

Datasets and Evaluation Measures. While MNIST-Rot12k [27], which is a rotated version of the original MNIST in any direction, is widely used for the classification task, there is no common dataset with this kind for the generic object detection task. Therefore, we constructed a new dataset[6] containing arbitrary rotation using MS COCO [33], called *COCO-Rot*, and evaluated the performance of our augmented feature pooling. Our *COCO-Rot* is composed of *COCO-Rot-train*

[6] The details of our dataset are described in our supplemental.



(a) mAP (b) AP_{50} (c) AP_{75}

Fig. 6. Performance on *COCO-Rot-val* by the proposed method with various number of augmentations ξ. The mAP, AP_{50}, and AP_{75} with MS COCO-train [33] and *COCO-Rot-train* as training data are shown, respectively.

and *COCO-Rot-val*, which were generated from the original MS COCO training and validation data, respectively. We automatically annotated the bounding box for *COCO-Rot-train* and *COCO-Rot-val* based on rotated ground truth segmentation mask instead of manual annotation. The numbers of images for *COCO-Rot-train* and *COCO-Rot-val* are 118K and 5K, respectively. For training, we used the original MS COCO-train [33] or *COCO-Rot-train* as training data, respectively. In addition, we also demonstrated our performance on two publicly available datasets, PASCAL VOC [17] and Synthetic Fruit Dataset [25]. We used MS COCO detection evaluation measures [33], i.e. the mean Average Precision (mAP), AP_{50}, and AP_{75}.

Implementation Details. We implemented our code based on MMDetection [5] with PyTorch [41]. The default training protocol in MMDetection [5] was employed unless otherwise noted. SGD was used for optimization, and the training schedule is 1x (i.e. 12 epochs with warmup and step decay, the learning rate is set to 2.0×10^{-2} to 2.0×10^{-4} for Faster-RCNN [45]). NVIDIA A100, P100 and K80 GPUs were used for our experiments. For evaluation of our framework and existing framework, we only trained the heads while fixing the feature extraction backbones. Unless otherwise noted, we used the pre-trained model using the original MS COCO [33] as initial weights for training. Resnet 50 was used in most of our experiments. We set the batch size to 16. The detailed training protocols are described in our supplementary material.

5.2 Effectiveness of Augmented Feature Pooling

We demonstrate the effectiveness of our proposed method in terms of the effectiveness by increasing the number of augmentations, the comparison with DA and TTA, the applicability to various backbones, and the effectiveness of our rotation set design using Faster-RCNN [45]. Applicability to various detection heads will be described in Sec. 5.3.

Improvement by increasing the number of augmentations. We first demonstrate the effectiveness of our proposed method by increasing the number of augmentations. Subscript ξ represents the number of augmentations as Ours_ξ. For example, ours with the four augmentations is denoted as Ours_4. We used three rotation sets with different numbers of augmentations ξ=4, 8, and 16,

Table 1. Performance of mAP on *COCO-Rot-val* by DA, TTA, and ours. Bold and italic indicate the best and second best results for each column, respectively. MS COCO-train and *COCO-Rot-train* are used as training data, respectively. Green and red characters show the increase or decrease in performance from vanilla.

(a) Original MS COCO-train.

Method	mAP
Vanilla	15.8
+ TTA_4	24.7 (+8.9)
+ DA_4	21.2 (+5.4)
+ DA_4+TTA_4	24.3 (+8.5)
Ours$_{16}$	21.3 (+5.5)
+ TTA_4	26.3 (+10.5)
+ DA_4	**26.7 (+10.9)**
+ DA_4 + TTA_4	*26.5 (+10.7)*

(b) *COCO-Rot-train.*

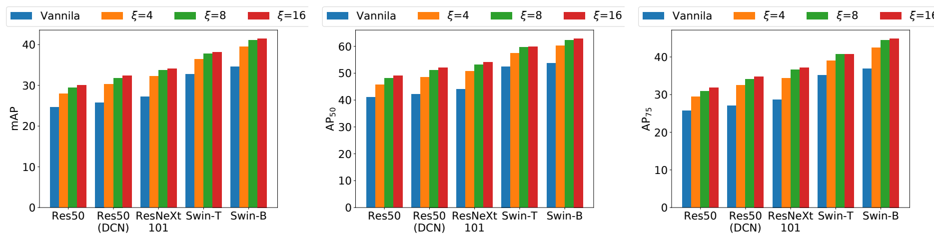
Method	mAP
Vanilla	24.6
+ TTA_4	27.6 (+3.0)
+ DA_4	24.8 (+0.2)
+ DA_4+TTA_4	27.8 (+3.2)
+ *Oracle DA_4*	24.8 (+0.2)
+ *Oracle DA_4*+TTA_4	27.7 (+3.1)
Ours$_{16}$	**30.0 (+5.4)**
+ TTA_4	**30.0(+5.4)**

where ξ is the number of augmentation for ours. Angles are assigned at equal intervals from all directions, depending on ξ. We only train the head of the detection network using *COCO-Rot-train* or the original MS COCO-train [33] while fixing the feature extraction backbones. The performance on *COCO-Rot-val* were evaluated. We also evaluated the performances of the original backbone feature extraction without our augmented feature pooling, which we call *vanilla* in the following. In vanilla, the original backbone is also frozen during training for their detection heads using *COCO-Rot-train* or original MS COCO-train, respectively.

Figure 6 shows mAP, AP_{50} and AP_{75} on *COCO-Rot-val* by the proposed method with various number of augmentations ξ. In our method, the performance is steadily improved as ξ is increased because we can avoid the loose bounding box problem. Note that naive DA, TTA and those combinations are only effective for $\xi = 4$ as shown in Sec. 3 (see Fig. 2 (d)). In the following, unless otherwise noted, for the number of augmentations in the following experiments, we set $\xi = 16$ for our proposed method, which was highest mAPs for our method. On the other hand, we fix $\xi = 4$ for DA, TTA as DA_4 and TTA_4 because DA, TTA and those combination are only effective for $\xi = 4$, i.e. Θ=$[-180, -90, 0, 90]$.

Comparison with DA and TTA. To demonstrate the superiority of our augmented feature pooling, we evaluated the performance of our method, DA, and TTA. For fair comparison, we also used *COCO-Rot-train* to train the heads for vanilla, DA, and TTA. We also compared our approach with a tightened bounding box using the instance segmentation mask label when perform rotation augmentation, called as *Oracle DA_4*. Table 1 shows mAP of our method, vanilla, DA, *Oracle DA*, and TTA. The values in parentheses are the increase (green) from mAP of the vanilla backbone feature extraction. We can clearly see that our proposed method with DA (Table 1 (a)) or TTA (Table 1 (b)) can achieve the highest mAP compared to naive DA and TTA [7]. Note that our proposed method also outperforms the tight box-based DA, called *Oracle DA*.

[7] As shown in our supplemental, AP_{50} and AP_{75} are also the highest in the proposed method as well as mAP.



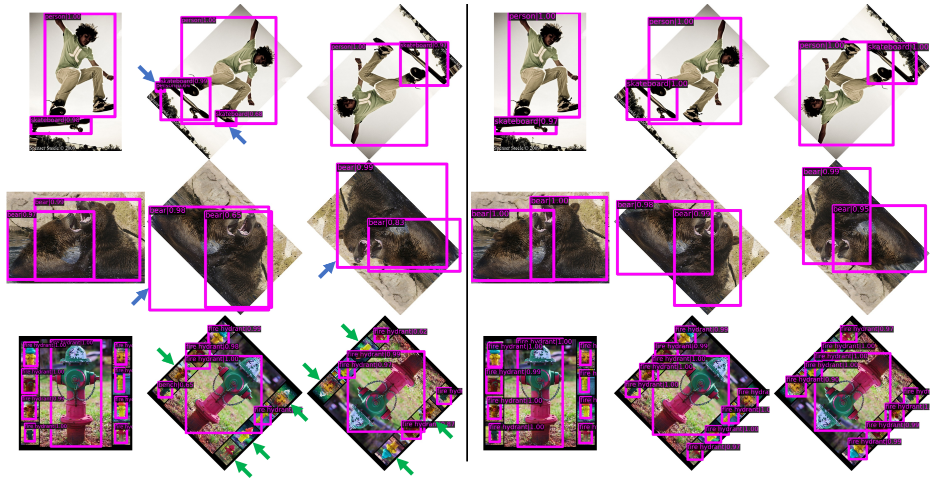
(a) mAP (b) AP$_{50}$ (c) AP$_{75}$

Fig. 7. Performance on *COCO-Rot-val* by the proposed method with various backbones. The proposed method is applicable to both CNN-based and transformer-based backbones. The robustness is improved by increasing the number of augmentations ξ.

This is because our augmented feature pooling can simultaneously solve both the rotation-bias problem of backbone feature extraction and the loosened bounding box problem as discussed in Sec. 3.

Applicability to Various Backbones. Our proposed method is effective for both CNN-based and transformer-based backbones. We evaluated the applicability of our framework using the following five major backbones with FPN [31], Resnet50 [21], Resnet50 [21] with DCN [12], Resnet101 [21], ResNeXt101 [56], Swin-T [37], and Swin-S [37]. The performance was evaluated on *COCO-Rot-val*. Again, the rotation set Θ was defined by uniformly assigning from all directions with equal intervals according to the number of augmentations ξ. Figure 7 shows mAP, AP$_{50}$ and AP$_{75}$ of our proposed framework with various numbers of augmentations and that of vanilla. Our framework substantially improves the performance for all backbones compared with the vanilla backbone feature extraction. We can see that the mAP, AP$_{50}$ and AP$_{75}$ are improved by increasing the number of augmentations ξ for all backbones. Note that our proposed method can further improve the performance of DCN [12] designed to compensate for the positional deformation. Our proposed method is applicable to such geometrical-transformation-based backbones. Figure 8 shows the visual comparisons between our proposed method and vanilla. There are many false positives in the vanilla backbone feature extraction (blue arrows in the first and the second rows) and false negatives (green arrows in the third row). Specifically, skateboards and people are falsely detected (blue arrows) in the first row, and fire hydrants (green arrows) can not be detected in the third row. In contrast, our proposed method can successfully detect those objects, even when using the same training dataset *COCO-Rot-train*. More visual comparisons are shown in our supplemental.

Effectiveness of Rotation Set Design. As described in Sec. 4.1, our framework can control whether to focus on the robustness to arbitrary angle range or a specific angle range by designing the rotation set Θ. To demonstrate this, we evaluated mAP, AP$_{50}$, and AP$_{75}$ for the three rotation set designs denoted by Set 1, Set 2, and Set 3 shown in Fig. 9. Here, Set 1 has only a single angle at 0 [deg], Set 2 has the five angles equally sampled among $\pm$45-degree range, and Set 3 has 16 angles equally sampled among $\pm$180-degree ranges. Figure 9

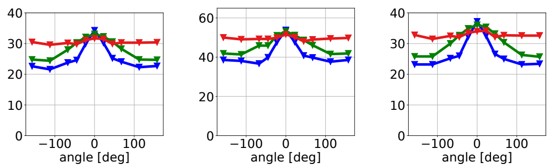


(a) Vanilla (original) (b) Vanilla (45[deg]) (c) Vanilla (225 [deg]) (d) **Ours** (original) (e) **Ours** (45[deg]) (f) **Ours** (225 [deg])

Fig. 8. Example results by our proposed method and vanilla using Faster-RCNN [45] with Swin-Transformer [37]. In the results of vanilla, the green and the blue arrows, i.e. ↖ and ↖, indicate **false negatives** and **false positives**, respectively. Contrary to vanilla backbone feature extraction, our proposed method can detect target objects with accurate bounding boxes for various rotation angles.

shows mAP, AP_{50}, and AP_{75} for each rotation angle for those three rotation set designs. Compared to Set 1 (blue line), mAP, AP_{50} and AP_{75} for Set 2 (green line) are improved in the wide-angle range centered on 0 [deg]. Furthermore, in Set 3 (red line), mAP, AP_{50}, and AP_{75} are improved on average for all rotation angles. From these results, we can see that our proposed method enables us to improve the robustness for arbitrary angles, and at the same time, it can also improve the robustness for a specific angle range by designing the rotation set.

Other Datasets. We compared the performance of our proposed method with the state-of-the-art method [25] focusing rotation augmentation for object detection using the PASCAL VOC and Synthetic Fruit datasets. For a fair comparison, the backbone and the detection head were both optimized as in [25]. We used ResNet50, which has the smallest expressive power of the backbone, in our implementation. Here, we set the rotation set Θ as seven angles sampled at equal intervals from a range of ± 15 [deg] as in [25]. Table 2 (a) shows AP_{50} and AP_{75} of the proposed and the existing methods. Note that the value for the existing method is taken from [25]. As shown in Table 2, our proposed method achieves substantially higher performance in both AP_{50} and AP_{75}. In contrast to [25], our framework can improve the robustness over a broader range by designing the rotation set Θ as mentioned in Sec. 4.1. In this sense, our framework is a more general and versatile framework that encompasses the existing method [25].



(a) mAP (b) AP_{50} (c) AP_{75}

Type	Rotation set Θ [deg]
Set1 ▼	0
Set2 ▼	-45, 22.5, 0, 22.5, 45
Set3 ▼	$22.5 \times n_\theta$, $n_\theta = \{-8, \cdots, 7\}$

Fig. 9. Performance comparisons with various rotation set. Our framework can control whether to focus on robust to arbitrary rotation or a specific angle range by designing the rotation set Θ.

Table 2. Result on other datasets. Bold indicates the most accurate methods. (a) Comparison of AP_{50} and AP_{75} on PASCAL VOC and Synthetic Fruit. (b) Results of AP_{50} on *PASCAL VOC-Rot*.

(a) PASCAL VOC and Synthetic Fruit

Datasets	PASCAL VOC [17]		Synth. Fruit [25]	
Methods	AP_{50}	AP_{75}	AP_{50}	AP_{75}
Ellipse+RU [25]	81.6	58.0	95.8	93.2
Ours	**89.6**	**69.4**	**96.7**	**93.6**

(b) *PASCAL VOC-Rot*

Method	AP_{50}	AP_{75}
Vanilla	64.9	35.6
+ TTA_4	72.3 (+7.4)	40.5 (+4.9)
Ours$_{16}$	77.3 (+12.4)	**45.7 (+10.1)**
+ TTA_4	**78.8 (+13.9)**	44.2 (+8.6)

In many practical scenarios, we cannot obtain an annotated segmented mask along object boundary due to the high cost of annotation. Even in such a case, the proposed method works better than naive TTA. To demonstrate this, we constructed a new dataset, called *PASCAL VOC-Rot*, by rotating the image and the originally annotated bounding box of the original PASCAL VOC in an arbitrary rotation. Table 2 (b) shows AP_{50} on *PASCAL VOC-Rot* [8]. Note that the bounding boxes for those evaluation and training datasets are loose because there are no segmented masks in PASCAL VOC. As shown in Table 2, our proposed method is relatively more effective than naive TTA.

5.3 Applicability to Modern Object Detection Architectures

The proposed method is applicable to various types of object detection networks including single-stage, two-stage, and transformer-based architectures. To demonstrate the versatility of our proposed method, the following widely used and state-of-the-art object detection networks were used for our evaluation: Faster-RCNN [45] (two-stage), Retinanet [32] (single-stage), YOLOF [6] (single-stage), FSAF [64] (anchor-free), ATSS [59] (anchor-free), and Deformable-DETR [65] (transformer-based).

Tables 3 (a) and (b) show mAP on *COCO-Rot-val* of our proposed method, vanilla with DA_4, our proposed method with TTA_4 and vanilla with TTA_4, respectively[9]. We can clearly see that our proposed method substantially improves the mAP for all the detection architectures than DA and TTA. Finally, we evaluated mAP, AP_{50}, and AP_{75} of our proposed method and vanilla with TTA_4 for each rotation angle as shown in Fig. 10. For mAP and AP_{75}, our proposed method outperforms vanilla with TTA_4 for almost all angles. In AP_{50}, the proposed method is comparable to vanilla with TTA_4 only in [65], and our method

[8] Note that, in PASCAL VOC, the standard evaluation metric is AP_{50}.
[9] We also show AP_{50} and AP_{75} in our supplementary material.

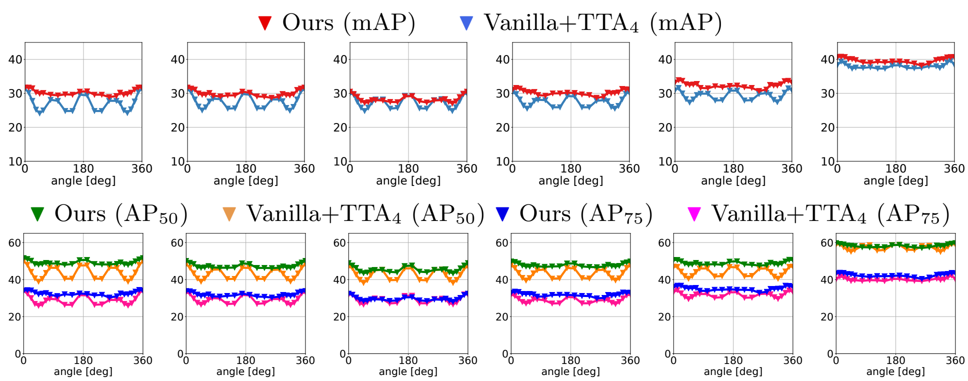
Table 3. Overall performance of mAP on *COCO-Rot-val*. Bold indicates the best result for each column. *COCO-Rot-train* is used for training.

(a) Our method and DA with various object detection networks.

Baseline	Backbone/Neck	Vanilla	Vanilla+DA_4	**$\mathbf{Ours}_{16}$**
Faster-RCNN [45]	ResNet50 w/ FPN	24.6	24.8 (+0.2)	**30.0 (+5.4)**
RetinaNet [32]	ResNet50 w/ FPN	24.1	24.3 (+0.2)	**29.6 (+5.5)**
FSAF [64]	ResNet50 w/ FPN	24.6	24.9 (+0.3)	**30.1 (+5.5)**
ATSS [59]	ResNet50 w/ FPN	26.6	27.0 (+0.4)	**32.2 (+5.6)**
YOLOF [6]	ResNet50	24.4	24.6 (+0.2)	**28.1 (+3.7)**
D-DETR (++ two-stage) [65]	ResNet50	35.9	37.3 (+1.4)	**39.5 (+3.6)**

(b) Our method and TTA with various object detection networks.

Baseline	Backbone/Neck	Vanilla	Vanilla+TTA_4	**$\mathbf{Ours}_{16}$+TTA_4**
Faster-RCNN [45]	ResNet50 w/ FPN	24.6	27.6 (+3.0)	**30.0 (+5.4)**
RetinaNet [32]	ResNet50 w/ FPN	24.1	27.3 (+3.2)	**29.5 (+5.4)**
FSAF [64]	ResNet50 w/ FPN	24.6	27.3 (+2.7)	**30.1 (+5.5)**
ATSS [59]	ResNet50 w/ FPN	26.6	29.3 (+2.7)	**32.0 (+5.4)**
YOLOF [6]	ResNet50	24.4	26.9 (+2.5)	**28.2 (+3.8)**
D-DETR (++ two-stage) [65]	ResNet50	35.9	37.6 (+1.7)	**39.2 (+3.3)**



(a) Faster [45] (b) Retina [32] (c) YOLOF [6] (d) FSAF [64] (e) ATSS [59] (f)D-DETR [65]

Fig. 10. Comparison of mAP, AP_{50} and AP_{75} for each rotation angle between our method and vanilla with TTA_4. The combination of various object detection with our method improves the robustness against rotation compared to vanilla with TTA_4.

outperforms vanilla with TTA_4 in other detection networks. These results show that our method is applicable to various detection networks.

6 Conclusions

We have proposed the rotation robust feature extraction framework using augmented feature pooling. The key is to integrate the augmented feature maps obtained from the rotated images before feeding it to the detection head without changing the original network architecture. We can obtain robustness against rotation using the proposed framework by freezing the backbone and fine-tuning detection head. Extensive experiments on three datasets demonstrated that our method improves the robustness of state-of-the-art algorithms. Unlike TTA and DA, the performance of the proposed method improves as the number of augmentations is increased.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation OSDI 16. pp. 265–283 (2016) 1
2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) p. 1–1 (2019). https://doi.org/10.1109/tpami.2019.2956516, http://dx.doi.org/10.1109/tpami.2019.2956516 3
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Eur. Conf. Comput. Vis. (ECCV) (2020) 1, 3
4. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019) 3
5. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019) 1, 9
6. Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., Sun, J.: You only look one-level feature. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2021) 3, 13, 14
7. Cheng, G., Han, J., Zhou, P., Xu, D.: Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. IEEE Trans. Image Process. **28**(1), 265–278 (2018) 3
8. Cheng, G., Zhou, P., Han, J.: Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 2884–2893 (2016) 3
9. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: Int. Conf. on Mach. Learn. (ICML) (2016) 3
10. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019) 1, 3
11. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: IEEE Conf. Comput. Vis. Pattern Recog. Workshop (CVPRW) (2020) 1, 3
12. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Int. Conf. Comput. Vis. (ICCV). pp. 764–773 (2017) 3, 11
13. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 1601–1610 (2021) 3
14. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning roi transformer for oriented object detection in aerial images. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 2849–2858 (2019) 4
15. Ding, J., Xue, N., Xia, G.S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Object detection in aerial images: A large-scale benchmark and challenges (2021) 4

16. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. (IJCV) **111**(1), 98–136 (2015) 1, 4
17. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. (IJCV) **88**(2), 303–338 (2010) 1, 4, 9, 13
18. Gao, S., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.H.: Res2net: A new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) (2019) 3
19. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 7036–7045 (2019) 3
20. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: A rotation-equivariant detector for aerial object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2021) 4
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2016) 1, 2, 3, 5, 7, 11
22. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. 2017 IEEE Int. Conf. Comput. Vis. (ICCV) (Oct 2017) 3
23. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Adv. Neural Inform. Process. Syst. (NeurIPS) (2015) 3
24. Jeon, Y., Kim, J.: Active convolution: Learning the shape of convolution for image classification. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 4201–4209 (2017) 3
25. Kalra, A., Stoppi, G., Brown, B., Agarwal, R., Kadambi, A.: Towards rotation invariance in object detection. In: Int. Conf. Comput. Vis. (ICCV) (2021) 3, 9, 12, 13
26. Laptev, D., Savinov, N., Buhmann, J.M., Pollefeys, M.: Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2016) 3
27. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: Int. Conf. on Mach. Learn. (ICML) (2007) 8
28. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Eur. Conf. Comput. Vis. (ECCV). pp. 765–781. Springer Verlag (2018) 3
29. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Adv. Neural Inform. Process. Syst. (NeurIPS) **33**, 21002–21012 (2020) 3
30. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. In: Adv. Neural Inform. Process. Syst. (NeurIPS) (2019) 1, 3
31. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 2117–2125 (2017) 6, 7, 11
32. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Int. Conf. Comput. Vis. (ICCV). pp. 2980–2988 (2017) 1, 3, 8, 13, 14
33. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Eur. Conf. Comput. Vis. (ECCV). pp. 740–755. Springer (2014) 1, 3, 4, 8, 9, 10

34. Liu, L., Pan, Z., Lei, B.: Learning a rotation invariant detector with rotatable bounding box. arXiv preprint arXiv:1711.09405 (2017) 4
35. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Eur. Conf. Comput. Vis. (ECCV) (2016) 3
36. Liu, Y., Wang, Y., Wang, S., Liang, T., Zhao, Q., Tang, Z., Ling, H.: Cbnet: A novel composite backbone network architecture for object detection. In: Proceedings of the AAAI Conf. on Artificial Intelligence (AAAI). pp. 11653–11660 (2020) 3
37. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. Int. Conf. Comput. Vis. (ICCV) (2021) 2, 3, 5, 6, 7, 11, 12
38. Marcos, D., Volpi, M., Komodakis, N., Tuia, D.: Rotation equivariant vector field networks. In: Int. Conf. Comput. Vis. (ICCV) (2017) 3
39. Pang, J., Chen, K., Li, Q., Xu, Z., Feng, H., Shi, J., Ouyang, W., Lin, D.: Towards balanced learning for instance recognition. Int. J. Comput. Vis. (IJCV) **129**(5), 1376–1393 (2021) 3
40. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019) 3
41. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inform. Process. Syst. (NeurIPS) **32**, 8026–8037 (2019) 1, 9
42. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 779–788 (2016) 3, 8
43. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 7263–7271 (2017) 3
44. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018) 3
45. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Adv. Neural Inform. Process. Syst. (NeurIPS) (2015) 1, 3, 4, 5, 8, 9, 12, 13, 14
46. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 1–48 (2019) 1
47. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Int. Conf. on Learn. Represent. (ICLR) (2015) 1, 3
48. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. arXiv preprint arXiv:1904.01355 (2019) 3
49. Vu, T., Jang, H., Pham, T.X., Yoo, C.D.: Cascade rpn: Delving into high-quality region proposal network with adaptive convolution. In: Adv. Neural Inform. Process. Syst. (NeurIPS) (2019) 3
50. Wang, Z., Tang, L., Liu, X., Yao, Z., Yi, S., Shao, J., Yan, J., Wang, S., Li, H., Wang, X.: Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In: Int. Conf. Comput. Vis. (ICCV). pp. 379–387 (2017) 3
51. Weng, X., Wu, S., Beainy, F., Kitani, K.M.: Rotational rectification network: Enabling pedestrian detection for mobile vision. In: Winter Conf. on Applications of Comput. Vis. (WACV). pp. 1084–1092. IEEE (2018) 3
52. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic networks: Deep translation and rotation equivariance. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2017) 3

53. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization for object detection. arXiv (2019) 3
54. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 1
55. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: The IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (June 2018) 4
56. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 1492–1500 (2017) 3, 7, 11
57. Xu, W., Wang, G., Sullivan, A., Zhang, Z.: Towards learning affine-invariant representations via data-efficient cnns. In: Winter Conf. on Applications of Comput. Vis. (WACV) (2020) 3
58. Yang, S., Pei, Z., Zhou, F., Wang, G.: Rotated faster r-cnn for oriented object detection in aerial images. In: Proc. of ICRSA (2020) 4
59. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. arXiv preprint arXiv:1912.02424 (2019) 3, 13, 14
60. Zhang, Z., Jiang, R., Mei, S., Zhang, S., Zhang, Y.: Rotation-invariant feature learning for object detection in vhr optical remote sensing images by double-net. IEEE Access (2019) 3
61. Zhang, Z., Chen, X., Liu, J., Zhou, K.: Rotated feature network for multi-orientation object detection. arXiv preprint arXiv:1903.09839 (2019) 3
62. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850 (2019) 3
63. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2017) 3
64. Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 840–849 (2019) 3, 13, 14
65. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: Int. Conf. on Learn. Represent. (2021), https://openreview.net/forum?id=gZ9hCDWe6ke 2, 3, 8, 13, 14