

# Towards Rotation Invariance in Object Detection

Agastya Kalra<sup>1</sup>, Guy Stoppi<sup>1</sup>, Bradley Brown<sup>1</sup>, Rishav Agarwal<sup>1</sup> and Achuta Kadambi<sup>1,2</sup>

<sup>1</sup>Akasha Imaging, Palo Alto CA

<sup>2</sup>UCLA, Los Angeles CA

{agastya, guy.stoppi, bradley.brown, rishav}@akasha.im, {achuta}@ee.ucla.edu

## Abstract

Rotation augmentations generally improve a model’s invariance/equivariance to rotation - except in object detection. In object detection the shape is not known, therefore rotation creates a label ambiguity. We show that the defacto method for bounding box label rotation, the Largest Box Method, creates very large labels, leading to poor performance and in many cases worse performance than using no rotation at all. We propose a new method of rotation augmentation that can be implemented in a few lines of code. First, we create a differentiable approximation of label accuracy and show that axis-aligning the bounding box around an ellipse is optimal. We then introduce Rotation Uncertainty (RU) Loss, allowing the model to adapt to the uncertainty of the labels. On five different datasets (including COCO, PascalVOC, and Transparent Object Bin Picking), this approach improves the rotational invariance of both one-stage and two-stage architectures when measured with AP, AP50, and AP75.

## 1. Introduction

It is desirable for object detectors to work when scenes are rotated. But there is a problem: methods like Convolutional Neural Networks (CNNs) may be scale and translation invariant but CNNs are not rotation invariant [15]. To overcome this problem, the training dataset can be expanded to include data at new rotation angles. This is known as *rotation augmentation*. In object detection, rotation augmentation can be abstracted as follows: given an original bounding box and any desired angle of rotation, how should we determine the axis-aligned rotated bounding box label? If the shape of the object is known, this is quite simple: we rotate the object shape and re-compute the bounding box. However in the case of object detection, the shape is unknown.

The prevailing folk wisdom in the community is to select a label large enough to completely overlap the rotated box [45]. In studying this problem, we find that

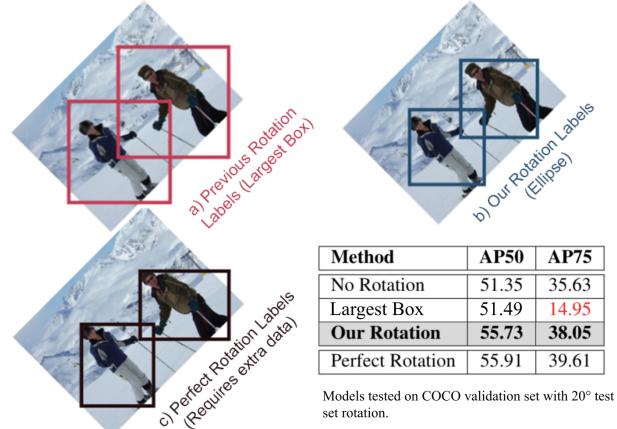


Figure 1: **A method is proposed to properly rotate a bounding box for rotation augmentation.** The previous solution of largest box is an overestimate of the perfect bounding box for a rotated scene. See table for how choice of rotation augmentation affects object detection performance.

this method may hurt performance, and on COCO [22], we find that *every other prior we tried is better*. Yet somehow this “**Largest Box**” method is very prevalent both in academia and in large scale object detection codebases [43, 4, 45, 37, 28, 26, 5, 7, 1, 17, 6, 40]. Indeed, recent analysis has found that largest box is only robust to about  $< 3^\circ$  [16] of rotation.

In this paper, we propose an advance on the largest box solution that achieves significantly better performance on five object detection datasets; while retaining the simplicity of a few lines of code implementation.

### 1.1. Contributions

In a nutshell, our solution has two aspects. First, we derive an **elliptical shape** prior from first principles to determine the rotated box label. We compare it to many other

novel priors and show this is optimal. Second, we introduce a novel **Rotation Uncertainty (RU) Loss** function, which allows the network to adapt the labels at higher rotations using priors from lower rotations based on label certainty. We demonstrate the effectiveness of this solution by both improving performance datasets where rotation is important such as Pascal VOC [14] and Transparent Object Bin Picking [18] and generalizing to novel test time rotations on MS COCO [22] (Figure 1).

## 1.2. Scope

Rotation data augmentation in object detection is not new. This paper is not about finding the best overall way to use a rotation data augmentation. For that - a brute force search or papers like AutoAugment [45] might be better examples. This paper focuses solely on methods to perform a rotation augmentation on axis-aligned bounding boxes. When implemented, these proposed modifications boil down to a few lines of code and leave little reason to use the current Largest Box method.

## 2. Related Work

**Data Augmentations** are an effective technique to boost the performance of object detection. Data augmentation increases the quantity and improves the diversity of data. Data augmentations are of two types. Photo-metric transforms modify the color channels such that the detector becomes invariant to change in lighting and color. Classical photometric techniques include adding Gaussian blur or adding colour jitters. Modern photo-metric augmentations like Cutout [11] and CutMix [44], randomly remove patches of the image. On the other hand, geometric transforms modify the image's geometry, making the detector invariant to position and orientation. Geometric modifications require corresponding changes to the labels as well. Geometric transforms are difficult to implement [5] and contribute more towards accuracy improvements [38]. We focus specifically on rotation augmentations and object detection.

**Rotation Augmentation in Object Detection** is currently done by the largest box method for major repositories (e.g. [5, 7, 1, 17, 6, 40]) and publications (e.g. [43, 4, 45, 37, 28, 26]) that do bounding box rotation for deep learning object detection. The largest box method does a great job guaranteeing containment, but at large angles, it severely overestimates the bounding box's size. Figure 2 shows an example of these over-sized bounding boxes. For that reason, FastAI [16] recommends rotation of no more than 3 degrees. Some recent work, such as AutoAugment [45, 37], use rotation as part of a complex learned data augmentation scheme. While learning rotation augmentation directly is interesting, it requires extensive computing resources. We

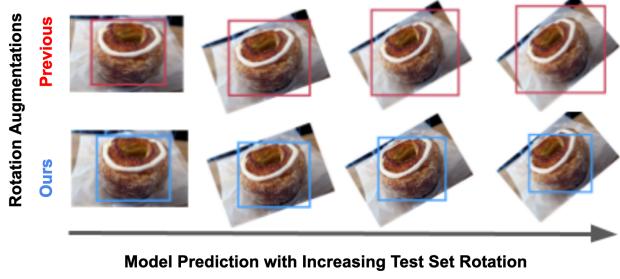


Figure 2: **Comparison between our method and other methods.** We show example predictions from models trained with each rotation augmentation above. Our method has comparable performance to using perfect segmentation labels without requiring the extra shape information.

seek to achieve the simplicity of the largest box, with performance improvements for larger angles.

**Oriented Bounding Boxes**, a sister of object detection, is the task of predicting non-axis aligned bounding boxes, also known as oriented bounding boxes. Several methods like [12, 8, 25], aim to achieve rotation invariance when predicting rotated bounding boxes. However, these methods already have labelled rotated boxes as input and do not end up with loose boxes when the input image is rotated. As this is a different task, it is out of the scope of this paper. Our paper focuses on axis-aligned bounding boxes only.

**Rotational Invariance** is an important problem to solve in object detection. Classical computer vision methods achieved rotational invariance by extracting features from images [41, 23, 24, 34]. With the rise of neural networks, newer methods attempt to modify the architecture to achieve rotation invariance [10, 9, 43]. These methods rotate the input images and add special layers that learn the object's orientation in the image. Our general-purpose method can again aid these methods as well.

## 3. Background

### 3.1. Rotation Augmentation for Object Detection

An image is parameterized by  $x$  and  $y$  coordinates. Suppose the image contains an object with shape  $S$ . Let  $S$  denote the **shape set** that describes all points in an object:

$$S = \{(x, y) \mid \forall (x, y) \in \text{object}\}. \quad (1)$$

In object detection, a bounding box is defined to be the tightest fitting axis-aligned box around a shape. Therefore a shape determines the coordinates of a bounding box,

$\mathbf{b} = [x_{min}, y_{min}, x_{max}, y_{max}]^T$ . Each of the four edges of the bounding box intersects at least one element of the shape set. Let the operator  $\mathcal{B}$  represent the perfect conversion of a shape to a bounding box:

$$\mathbf{b} \triangleq \mathcal{B}(S). \quad (2)$$

The operator  $\mathcal{B}$  extracts the bounding box  $\mathbf{b}$  (tightest fitting axis aligned box) for  $S$  by taking the minimum/maximum  $(x, y)$  coordinates of the shape. As  $S$  is not unique, the same bounding box can be generated by many shapes. For example, a square of side length  $d$  and a circle of diameter  $d$  are just two of many unique shapes that generate the same bounding box. More formally, let  $V_{\mathbf{b}}$  denote the set of shapes that could possibly generate a bounding box  $\mathbf{b}$ , such that

$$V_{\mathbf{b}} = \{S_i \mid \mathcal{B}(S_i) = \mathbf{b}, \forall S_i \in \mathcal{P}\}. \quad (3)$$

Where  $\mathcal{P}$  is the dataset specific distribution of shapes. Let us consider the problem of rotation augmentation where an image and corresponding box label  $\mathbf{b}$  is rotated by angle  $\theta$ . If the shape of the object is known, then we can rotate the original shape by angle  $\theta$  using a rotation operator:  $\mathcal{R}^\theta(S)$ . In analogy to Equation 2, we can then use the perfect method to obtain an axis-aligned bounding box for the rotated image as:

$$\mathbf{b}_{\text{perfect}}^\theta \triangleq \mathcal{B}(\mathcal{R}^\theta(S_{\text{perfect}})), \quad (4)$$

We call this method **perfect labels**, where  $S_{\text{perfect}}$  is the actual shape of the object for a given bounding box  $\mathbf{b}$ . However, this requires shape labels, which are not available for object detection. In object detection, humans label boxes by implicitly segmenting the shape. Without knowing the shape labels, any shape  $S \in V_{\mathbf{b}}$  could be  $S_{\text{perfect}}$ , leading to many possible boxes  $\mathbf{b}^\theta$ . This paper seeks a method to estimate the rotated bounding box when we do not know the shape. We are only provided with the original bounding box  $\mathbf{b}$ , which we will hereafter write as  $\mathbf{b}^0$  by making explicit that  $\theta = 0$ . The problem statement follows.

**Problem Statement:** Given only an input bounding box  $\mathbf{b}^0$  and an angle  $\theta$  by which the image should be rotated, find the axis aligned bounding box  $\hat{\mathbf{b}}^\theta$  that: (1) has high IoU with  $\mathbf{b}_{\text{perfect}}^\theta$ ; and (2) improves model performance on rotated versions of vision datasets.

### 3.2. Largest Valid Box Method

Rotation augmentation without shape knowledge is not a new problem statement. The *de facto* method in the object detection community for determining the bounding

box post-rotation with no shape priors is the **largest box** method. The largest box method is *extremely* prevalent (e.g. [28, 26, 5, 7, 1, 17, 6, 40, 20, 19, 27, 33, 36, 39]). Just like our proposed method, the largest box takes only the original bounding box  $\mathbf{b}^0$  and  $\theta$  as input. From Equation 3 it is clear that several shapes could define  $\mathbf{b}^0$ . This creates an *ambiguity problem*. The largest box method chooses the single largest of these possible shapes in area,  $S_{\text{largest}}$ . This shape is simply the box itself (Table 1). Treating this as the object shape, Equation 4 can be adapted to obtain

$$\mathbf{b}_{\text{largest}}^\theta \triangleq \mathcal{B}(\mathcal{R}^\theta(S_{\text{largest}})). \quad (5)$$

The benefit of this method is that it produces a box that is *guaranteed* to contain the original object [45], and it is easy to implement. The downside is that the method produces oversized boxes [16, 45, 35, 2, 29, 3], and if used generously, hurts performance more than it helps (Table 9). Surprisingly, to our best knowledge, including personal communication with practitioners and posts on internet forums, no alternatives have been adopted. We hope our method will change that.

## 4. Proposed Solution

We now describe our solution to the problem: given  $\mathbf{b}^0$  and desired rotation angle  $\theta$ , find  $\hat{\mathbf{b}}^\theta$ . In a nutshell, our solution estimates a rotated bounded box by assuming the original shape is an ellipse (Table 1, Figure 3) and rotating accordingly (Section 4.1). We then adapt the loss function to account for error in the labels (Section 4.3).

### 4.1. Ellipse Method

In this section, we first derive the ellipse assumption from first principles by trying to find the shape that is most likely to have high overlap with potential ground truth boxes. Then we discuss the implementation and intuition of the ellipse method. Finally, we mention other novel methods we developed.

#### 4.1.1 Ellipse from Maximizing Expect IoU

We start with a simple assumption: the optimal method for determining a bounding box post-rotation augmentation should maximize label accuracy, which in the case of object detection is measured in IoU.

We define  $\hat{\mathbf{b}}^\theta$  as the optimal rotated bounding box. We are provided the input angle  $\theta$  and box  $\mathbf{b}^0$ . From Equation 3, this box could have been generated from any number of shapes:  $V_{\mathbf{b}^0} = \{S_i\}_{i=1,2,\dots,N}$ . For each shape we can use the “perfect method” from Equation 4 to obtain a potential rotated bounding box. Since multiple shapes can lead to the same rotated box, we obtain  $M \leq N$  possible bounding boxes, which we write as the set:

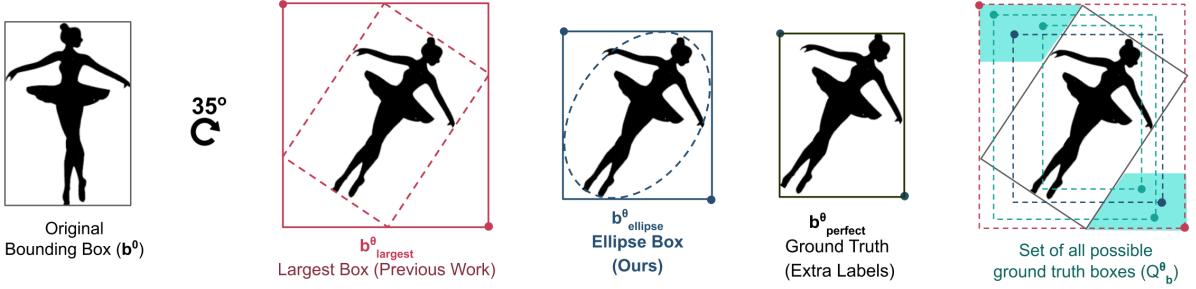


Figure 3: Our Ellipse method leads to good initial training labels while the Largest Box overestimates the labels. From left to right: (1) The original bounding box prior to rotation. (2) The oversized Largest Box estimate of the ground truth label post rotation. (3) The tighter Ellipse estimate (Section 4.1). (4) The actual ground truth which we get from segmentation shape labels. (5) The set of all possible ground truth boxes given a rotation and an initial box.

$$Q_{b^0}^\theta = \text{unique}\{\mathbf{b}_j^\theta = \mathcal{B}(\mathcal{R}^\theta(S_i)), \forall S_i \in V_{b^0}\} \quad (6)$$

$$= \{\mathbf{b}_j^\theta\}_{j=1,\dots,M}. \quad (7)$$

Hereafter, and without loss of generality, the paper will assume that  $b^0$  is the input allowing notation to be simplified:

$$Q^\theta \triangleq Q_{b^0}^\theta. \quad (8)$$

Now the task becomes to pick the “best” of the  $M$  possible bounding boxes in  $Q^\theta$ . Recall that the de facto solution is to choose the largest box in  $Q^\theta$ . This largest box is guaranteed to contain the object. However, optimizing for containment does not seem like a good choice to directly address the metric of AP because AP uses IoU to determine true positives, not containment. A more relevant goal for object detection is to select a box that maximizes:

$$\hat{\mathbf{b}}^\theta = \underset{\hat{\mathbf{b}}^\theta \in Q^\theta}{\text{argmax}} \text{IoU}(\hat{\mathbf{b}}^\theta, \mathbf{b}_{\text{perfect}}^\theta). \quad (9)$$

In which case  $\hat{\mathbf{b}}^\theta = \mathbf{b}_{\text{perfect}}^\theta$ . Of course, we are not given  $\mathbf{b}_{\text{perfect}}^\theta$ . So for the moment, let us assume any of the bounding boxes in  $Q^\theta$  has an equal chance of being the perfect box. Then, it would make sense to optimize over:

$$\hat{\mathbf{b}}^\theta = \underset{\hat{\mathbf{b}}^\theta \in Q^\theta}{\text{argmax}} \text{Avg}\{\text{IoU}(\hat{\mathbf{b}}^\theta, \mathbf{b}_j^\theta) \mid \forall \mathbf{b}_j^\theta \in Q^\theta\}. \quad (10)$$

Now, let us break the assumption that each candidate box in  $Q^\theta$  is equally likely to be the perfect box. Indeed, we know that many shapes can produce the same box (since  $M \leq N$ ), so certain boxes are more likely than others. For example, the only shape that can produce the largest box is the original box itself, whereas other rotated boxes can

Method	Shape Definition
$S_{\text{largest}}$	$\{(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_1, y_2)\}$
$S_{\text{ellipse}} \text{ (Ours)}$	$\left\{ (x, y) \mid \frac{(x - x_c)^2}{(b_w^0/2)^2} + \frac{(y - y_c)^2}{(b_H^0/2)^2} = 1 \right\}$

Table 1: Our method can be compared with the Largest Box in the shape domain. The implementation difference is one line of code.

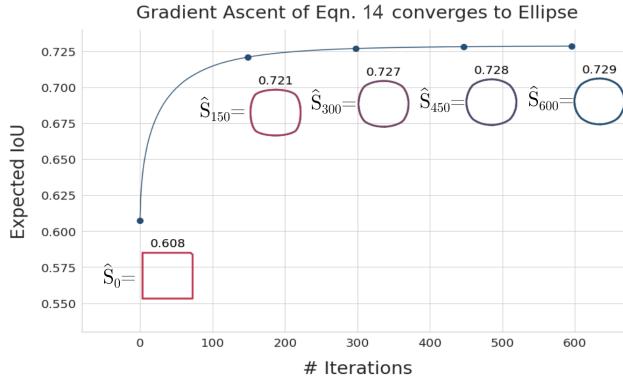
be generated by multiple object shapes in the dataset. Denote  $p(\mathbf{b}_j^\theta)$  as the probability that box  $\mathbf{b}_j^\theta = \mathbf{b}_{\text{perfect}}^\theta$ . Then Equation 10 can be reformulated as:

$$\hat{\mathbf{b}}^\theta = \underset{\hat{\mathbf{b}}^\theta \in Q^\theta}{\text{argmax}} \sum_{i=1}^M p(\mathbf{b}_j^\theta) \text{IoU}(\hat{\mathbf{b}}^\theta, \mathbf{b}_j^\theta). \quad (11)$$

Readers may recognize this equation as being analogous to an expectation. We refer to this in the paper as the **Expected IoU**. The expected IoU is not directly tractable: we do not know  $p(\mathbf{b}_j^\theta)$  *a priori*. However, if we can sample  $K$  random shapes from a dataset distribution over shapes  $S_k \sim \mathcal{P}$  where  $\mathbf{b}_k^\theta = \mathcal{B}(\mathcal{R}^\theta(S_k))$  we get the following optimization objective:

$$\hat{\mathbf{b}}^\theta = \underset{\hat{\mathbf{b}}^\theta \in Q^\theta}{\text{argmax}} \frac{1}{K} \sum_{k=1}^K \text{IoU}(\hat{\mathbf{b}}^\theta, \mathbf{b}_k^\theta). \quad (12)$$

Since all object detection datasets do not have shape labels, we sample  $\mathcal{P}$  by generating random shapes that touch each side of  $b^0$  once. This way we are not dataset-specific. We analyze using COCO shapes in the supplement and show the performance is extremely similar. The above equation is fully differentiable, and so we can solve with gradient ascent. The problem here is that we would then have to solve this equation for every  $\theta$  and every box  $b$ , and this is not practical. Therefore we generalize this further to a canonical shape.



Label Method	$S_{\text{largest}}$	$S_{\text{ellipse}} (\text{Ours})$	$\hat{S} (\text{Ours})$
Expected IoU	60.8	<b>72.9</b>	<b>72.9</b>

Figure 4: **The optimal shape to maximize expected IoU with potential ground truth boxes converges to an ellipse.** Curve showing the progression of gradient ascent starting from the largest box and converging to an elliptical shape. The final converged expected IoU for  $\hat{S}$  matches that of the largest inscribed ellipse  $S_{\text{ellipse}}$ .

**Optimizing Equation 12 via a canonical shape:** Instead of solving Equation 12 for every possible combination of  $b^0$  and  $\theta$ , we attempt to find a shape that is optimal across different input bounding boxes. This way, we could solve for some best shape  $\hat{S} \in V_{b^0}$ , and solve for  $\hat{\mathbf{b}}^\theta$  as follows:

$$\hat{\mathbf{b}}^\theta \triangleq \mathcal{B}(\mathcal{R}^\theta(\hat{S})), \quad (13)$$

To obtain the likely shape, we combine Equations 12 and 13 to optimize the quantity:

$$\hat{S} = \underset{S \in V_{b^0}}{\operatorname{argmax}} \sum_{\theta} \frac{1}{K} \sum_{k=1}^K [IoU(\mathcal{B}(\mathcal{R}^\theta(S)), \mathbf{b}_k^\theta)]. \quad (14)$$

Note that we now optimize over all rotation angles and aspect ratios simultaneously. This adds enough constraints to find a unique shape. The goal is to find the shape  $\hat{S}$  that produces an augmented bounding box that has high IoU with likely ground truth boxes. Since  $\mathcal{R}$  and  $\mathcal{B}$  are differentiable operators, Equation 14 can be optimized through gradient ascent to solve for  $\hat{S}$ . We provide details and pseudo-code and some analysis in the supplement. The stable solution found by gradient ascent is that of an elliptical shape. We show the progression of gradient ascent in Figure 4 from the largest box shape to a circle. If we change the aspect ratio, it simply converges to the largest inscribed ellipse. Also in Figure 4, we show the Expected IoU for the Largest Box shape is much lower than the Ellipse, and in Figure 5 we show that the resulting AP of the Ellipse labels is much better. The elliptical solution is similar to the optimized shape

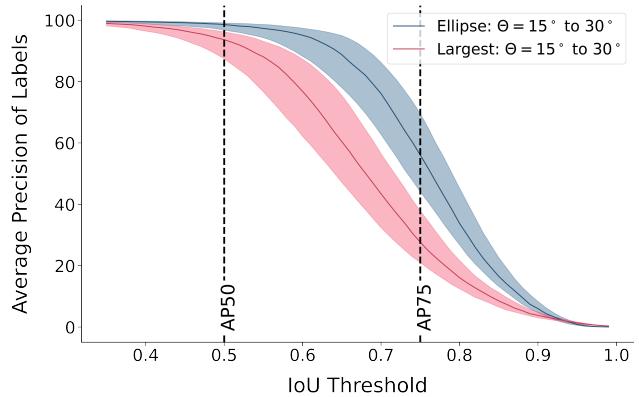


Figure 5: **Comparing Label AP (assuming uniform confidence) at different IOU thresholds for both methods at a  $15^\circ - 30^\circ$  rotation augmentation.** Ours is significantly better at AP50 and AP75.

for various tested distributions of  $\mathcal{P}$ , including the random model described in the previous paragraph.

#### 4.1.2 The Ellipse Method

When we model the shape as an ellipse, we can find the estimated bounding box as:

$$\hat{\mathbf{b}}^\theta = \mathcal{B}(\mathcal{R}^\theta(S_{\text{ellipse}})), \quad (15)$$

where  $S_{\text{ellipse}}$  is the largest inscribed ellipse inside  $b^0$ , expressed as:

$$S_{\text{ellipse}} = \left\{ (x, y) \mid \frac{(x - x_c)^2}{(b_W^0/2)^2} + \frac{(y - y_c)^2}{(b_H^0/2)^2} = 1 \right\}, \quad (16)$$

where  $(x_c, y_c)$  is the location of the center of  $b^0$  and  $b_W^0, b_H^0$  are the width and height of  $b^0$  respectively.

This equation is **fast, simple to implement, and high-performing on modern vision datasets**. The elliptical approximation can be implemented in the same line of code as the largest box method (cf. Appendix A), yet it greatly improves performance. We see in Figure 5 the ellipse labels are far more accurate than the largest box labels. However, one disadvantage with the proposed elliptical box method is that the elliptical box can underestimate the object size or aspect ratio. This still causes some noise in the labels, especially at large rotations. We mitigate this by allowing the model to adapt labels at higher rotations based on priors from lower rotations in Section 4.3.

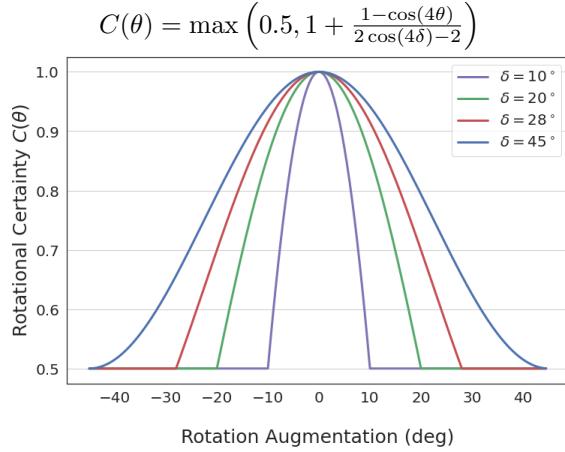


Figure 6: **Rotational certainty** used by RU Loss as a function of theta plotted for different hyper-parameters  $\delta$  where  $\delta$  is the angle at which  $C(\theta) = 0.5$ .

## 4.2. Other Methods

We do not limit our analysis to the ellipse method. To perform a complete study we came up with an additional 4 methods. To conserve space, full details and results of these novel methods are available in the supplement Appendix B, we provide a quick summary for these methods here.

- Scaled Octagon:* We use an octagon with a scaling factor ( $s$ ) to interpolate between the largest box shape and a diamond shape.
- Random Boxes:* We sample random valid boxes and use those as ground truth labels.
- RotIoU:* We select the label that has the maximum IoU with the rotated ground truth box rather than the expected axis-aligned ground truth box.
- COCO Shape:* Rather than using random shapes for the optimization, we use the shapes from the COCO dataset. We keep results from this to the supplement since the performance between this and the ellipse method is negligible and we want this paper to be dataset independent and easy to implement.

## 4.3. Rotation Uncertainty Loss

As shown in Figure 4 the expected IoU with random shapes is 72.9. This means attaining good performance at the higher APs, like AP75, will be very difficult using just these labels. To tackle this problem, we create a custom loss function that adapts the regression loss to account for the uncertainty of the rotation. The idea is simple - if we are uncertain of the label, we turn off the regression loss if the model is *close enough*. The labels are more uncertain

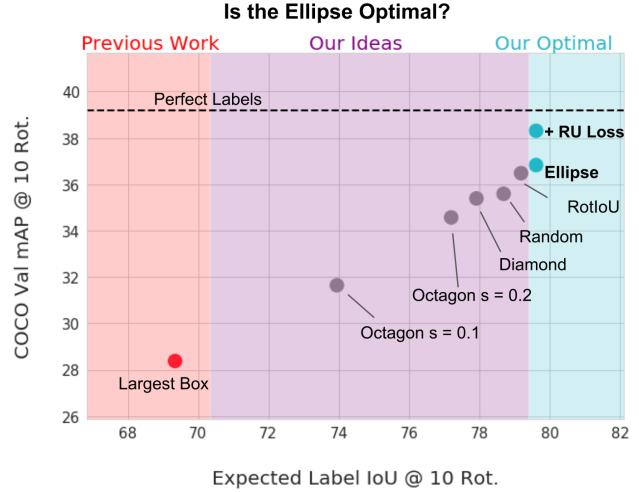


Figure 7: **The Expected IoU of a rotation method is heavily correlated with performance.** Our final Ellipse is optimal for both.

COCO val2017 Ablations	0°	10°	20°	30°
(a) Previous method				
Largest Box(e.g. [4, 45, 37])	35.20	28.37	22.34	18.47
(b) Our Rotation Label Methods (Section 4.2)				
Random	37.39	35.59	32.22	28.33
Octagon $s = 0.1$	35.82	31.64	27.16	23.54
Octagon $s = 0.2$	36.52	34.57	31.65	28.15
Octagon $s = 0.5$ (Diamond)	<b>38.36</b>	35.39	28.76	22.92
RotIoU	<b>38.32</b>	36.48	32.68	28.94
<b>Ellipse</b> (Section 4.1.2)	<b>38.21</b>	<b>36.83</b>	<b>33.59</b>	<b>29.95</b>
(c) With Our Loss (Section 4.3)				
Ellipse + RU Loss $\delta = 45^\circ$	38.54	37.45	34.56	31.26
Ellipse + RU Loss $\delta = 30^\circ$	39.09	37.99	35.45	32.25
Ellipse + RU Loss $\delta = 15^\circ$	<b>39.14</b>	<b>38.19</b>	35.78	32.50
<b>Ellipse + RU Loss <math>\delta = 10^\circ</math> (Final)</b>	<b>39.33</b>	<b>38.31</b>	<b>36.00</b>	<b>32.72</b>

Table 2: **The AP at different test rotations on the COCO val2017 set for different methods.** (a) The previous method of largest box leads to the worst performance - every other idea we had was better. (b) The Ellipse is the best of all the label generation methods. (c) Our RU Loss with  $\delta = 10^\circ$  leads to the best AP across all rotations and therefore we use this as our final method. *Note: We bold within 0.2 of best result.*

as the rotation approaches  $45^\circ, 135^\circ, 225^\circ, 315^\circ$ . and perfectly certain at  $0^\circ, 90^\circ, 180^\circ$  and  $270^\circ$ . We formalize on the concept of certainty (Figure 6) as a function of  $\theta$ :

$$\alpha = 2 \cos(4\delta), \quad (17)$$

$$C(\theta) = 1 + \frac{1}{\alpha - 2}(1 - \cos(4\theta)). \quad (18)$$

This function maps a rotation  $\theta$  to an IoU threshold  $C(\theta)$ . We use this IoU threshold to serve as an indicator for applying regression loss. If the predicted box is greater than

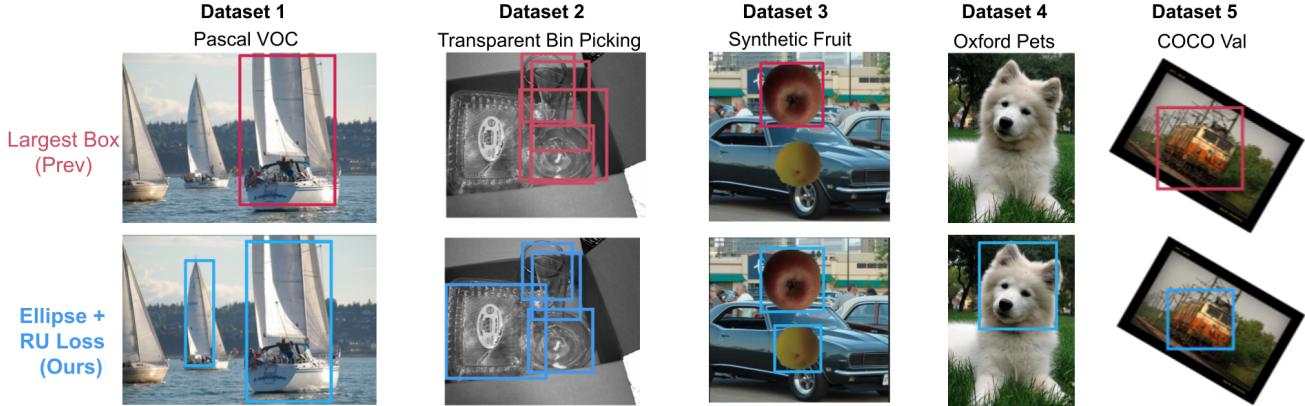


Figure 8: **Example bounding box predictions for Largest Box model (top row) and Ellipse model (bottom row) for all 5 datasets.** We can see that ours produces tighter bounding boxes overall.

Datasets (At 0° Test Rotation)	Pascal VOC [14]			Transparent Bin Picking [18]			Synthetic Fruit Dataset [13]			Oxford Pets [30]		
	AP	AP50	AP75									
No Rotation	51.94	80.91	56.54	48.53	79.14	54.3	84.3	95.07	92.6	80.70	92.80	88.76
Largest Box(e.g. [4, 45, 37]) (relative improvement)	50.23 -3.29%	81.31 0.49%	54.3 -3.96%	37.49 -22.7%	79.09 -0.06%	28.45 -47.6%	83.47 -0.98%	95.05 -0.02%	92.24 -0.39%	79.54 -1.43%	94.20 1.56%	90.03 1.43%
Ellipse + RU Loss (Ours) (relative improvement)	<b>52.89</b> <b>1.84%</b>	<b>81.57</b> <b>0.82%</b>	<b>57.97</b> <b>2.53%</b>	<b>50.36</b> <b>3.78%</b>	<b>81.78</b> <b>3.35%</b>	<b>56.76</b> <b>4.53%</b>	<b>84.83</b> <b>1.05%</b>	<b>95.83</b> <b>0.80%</b>	<b>93.17</b> <b>0.62%</b>	<b>81.28</b> <b>0.72%</b>	<b>94.37</b> <b>1.69%</b>	<b>91.09</b> <b>2.63%</b>

Table 3: **Across four separate datasets we show that our method of rotation augmentation leads to an improved performance where the previous method hurts performance.** Especially in the case of transparent object bin picking where the largest box is almost 50% worse and ours is 4.5% better AP75.

$\max(0.5, C(\theta))$ , it uses the regression loss, otherwise, it does not and assumes the model’s prediction is correct. We parameterize  $C$  with  $\delta$ .  $\delta$  is the angle at which  $C(\theta) = 0.5$ . We visualize  $C$  in Figure 6. We bound it by 0.5 since that is the threshold for anchor-matching in standard object detection architectures [21].

This function allows the model to take the priors it learns at the confident rotations and apply them to the higher rotations, preventing it from overfitting to poor labels. We show in Table 2.

## 5. Results

### 5.1. Setup

Our hardware setup contains only a single P100 GPU for training, and all our code is implemented in Detectron2 [42] with Pytorch [31]. We use the default training pipeline for both Faster-RCNN [32] and RetinaNet [21]. We conduct most of our experiments on the standard COCO benchmark since it contains a variety of objects with many different shapes - making it a challenging test set.

**Training** Since we have only a single GPU, we can only fit a batch size of 3. To account for this, we increase the training time by around 5x from the default configurations. This allows us to match online available pre-trained baselines for RetinaNet [21] and Faster-RCNN [32]. Since most

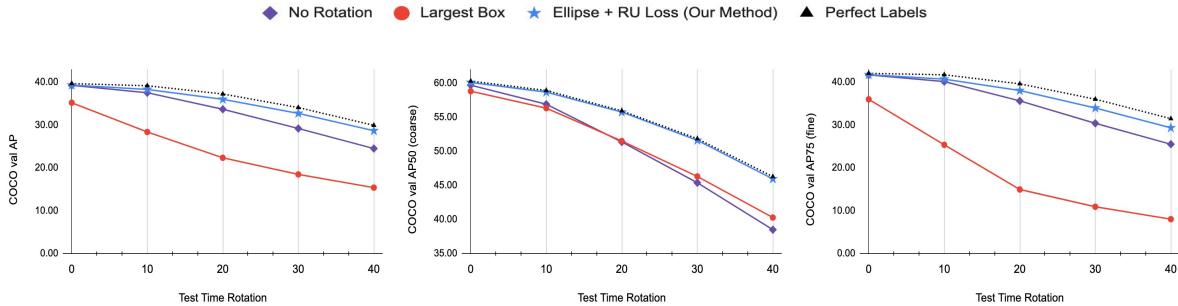
datasets are right-side-up images, we train with a normal distribution with a mean of 0 and a standard deviation of 15 degrees for all experiments. Since this paper aims to find the optimal rotation augmentation method, not the strategy for applying rotation augmentation, we do not try other combinations. This may be left for future work.

**Testing:** For all datasets except COCO we do not have complete segmentation labels, so we only test on the standard test set ( $0^\circ$ ). For COCO we generate our test set by taking the COCO val2017 set and rotating it from  $0^\circ$ - $40^\circ$  to simulate out-of-distribution rotations. We then bucket these rotations in intervals of 10 and evaluate using segmentation labels to generate ground truth. We leave COCO results for Faster-RCNN to supplement because they are similar to the results for RetinaNet shown below.

### 5.2. Ablation Studies

In Figure 7 and the accompanying table, we conduct a thorough ablation study on both the method for choosing the label and the impact of the RU Loss function.

**Justifying EIoU Optimization:** In Section 4.1.1 we assumed that the optimal method for label rotation should maximize label accuracy, which we approximated as Expected IoU (Eq. 14). In Figure 7 we demonstrate a strong correlation between Expected IoU and performance



COCO val2017 Results	AP				AP50				AP75			
	0°	10°	20°	30°	0°	10°	20°	30°	0°	10°	20°	30°
No Rotation	39.26	37.54	33.68	29.19	59.68	56.88	51.35	45.37	41.69	40.14	35.63	30.39
Largest Box (e.g. [4, 45, 37]) (relative improvement)	35.20	28.37	22.34	18.47	58.79	56.31	51.49	46.30	36.00	25.37	14.95	10.91
-10.3% -24.4% -33.7% -36.7%	-1.48% -1.01% 0.27% 2.04%	-13.6% -36.8% -58.1% -64.1%										
Ellipse + RU Loss (Ours) (relative improvement)	<b>39.33</b> <b>0.17%</b>	<b>38.31</b> <b>2.05%</b>	<b>36.00</b> <b>6.88%</b>	<b>32.72</b> <b>12.1%</b>	<b>60.08</b> <b>0.67%</b>	<b>58.66</b> <b>3.12%</b>	<b>55.73</b> <b>8.54%</b>	<b>51.60</b> <b>13.7%</b>	<b>41.74</b> <b>0.13%</b>	<b>40.71</b> <b>1.42%</b>	<b>38.05</b> <b>6.79%</b>	<b>33.97</b> <b>11.8%</b>
Perfect Labels (relative improvement)	39.66 1.02%	39.17 4.36%	37.24 10.6%	34.08 16.7%	60.28 1.00%	58.89 3.53%	55.91 8.88%	51.83 14.2%	42.05 0.88%	41.73 3.94%	39.61 11.2%	36.02 18.5%

Figure 9: **Our method of Ellipse + RU Loss performs close to perfect labels for AP50** and performs better than No Rotation and Largest Box for AP and AP75 - demonstrating the first reliable rotation augmentation without shape labels.

on COCO at  $10^\circ$  across all methods, proving the effectiveness of our first principles derivation. We see similar correlations at other angles as well.

**Justifying the Ellipse:** In Section 4.1 we introduced many potential methods for rotating a box label. In the ablation Table 2b we show that the Ellipse leads to the best performance across all rotations except  $0^\circ$  where it is within a small noise tolerance. It is also important to note that all methods we tried perform significantly better than the Largest Box - showing the importance of fixing this issue.

**RU Loss Ablation:** In Section 4.3 we introduce a hyper-parameter  $\delta$  in our final method. We ablate over that in Table 2 and demonstrate that  $10^\circ$  is optimal. We found this to be true on COCO, however, on simpler datasets we use larger values of  $\delta$  for optimal performance.

### 5.3. Overall Performance

Our best performing method consists of using both Ellipse-based label rotation and RU Loss. In this section, we show it leads to much better performance across multiple datasets and approximates segmentation-based rotation augmentations on COCO.

#### 5.3.1 Object Detection Datasets

In Table 3, we provide four datasets where our method of rotation augmentation improves performance while the previous one (Largest Box) hurts performance. We notice this

to be especially bad at higher APs, such as AP75. The gap is also larger in complex datasets such as transparent object bin picking where the largest box reduces performance by almost 50% and ours increases it by 4.5%.

#### 5.3.2 Generalizing to new Rotation Angles

Our method significantly outperforms the original largest box method and also outperforms not using rotation for AP, AP50, and AP75 across all new angles from  $[0^\circ - 30^\circ]$  on COCO in Figure 1 and Figure 9. In the case of AP50, we show very similar improvements compared to using segmentation-based labels. This is a huge improvement since the largest box method hurts rotation performance.

## 6. Conclusion

The widespread Largest Box method (e.g. [43, 4, 45, 37, 28, 26]) is based on the folk wisdom of maximizing overlap. Instead, we show that by maximizing Expected IoU and accounting for label certainty in the loss, we can **completely match** the performance of perfect “segmentation-based” labels at AP50 while also achieving **significant gains** for AP and AP75. These results represent a step toward achieving rotation invariance for object detection models, while adding only a few lines of complexity to object detection codebases.

**Acknowledgements:** We thank Yuri Boykov, Tomas Gerlich, Abhijit Ghosh, Olga Veksler and Kartik Venkataraman for their helpful discussions and edits to improve the paper.

## References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.
- [2] Albumentations-Team. Why does the bounding box become so loose after rotation data aug? is there a better way? · issue 746 · albumentations-team/albumentations. <https://github.com/albumentations-team/albumentations/issues/746>.
- [3] Aleju. Make the bounding box more tighten for the object after image rotation · issue 90 · aleju/imgaug. <https://github.com/aleju/imgaug/issues/90>.
- [4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [5] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.
- [6] Ángela Casado-García, César Domínguez, Manuel García-Domínguez, Jónathan Heras, Adrián Inés, Eloy Mata, and Vico Pascual. Clodsa: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks. *BMC bioinformatics*, 20(1):323, 2019.
- [7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mm-lab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [8] Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2018.
- [9] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016.
- [10] Gong Cheng, Peicheng Zhou, and Junwei Han. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2884–2893, 2016.
- [11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [12] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [13] Brad Dwyer. How to create a synthetic dataset for computer vision. <https://blog.roboflow.com/how-to-create-a-synthetic-dataset-for-computer-vision/>, Aug 2020.
- [14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [16] Jeremy Howard. Lesson 9: Deep learning part 2 2018 - multi-object detection - fast ai. <https://youtu.be/0frKXR-2PY?t=554>.
- [17] Alexander B Jung, K Wada, J Crall, S Tanaka, J Graving, S Yadav, J Banerjee, G Vecsei, A Kraft, J Borovec, et al. imgaug, 2018.
- [18] Agastya Kalra, Vage Taamazyan1, Sreeth Krishna Raol, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [19] Ayoosh Kathuria. Data augmentation for bounding boxes: Rethinking image transforms for object detection. <https://www.kdnuggets.com/2018/09/data-augmentation-bounding-boxes-image-transforms.html>.
- [20] Ayoosh Kathuria. Data augmentation for object detection: How to rotate bounding boxes. <https://blog.paperspace.com/data-augmentation-for-object-detection-rotation-and-shearing/>, Sep 2018.
- [21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [22] 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 *European conference on computer vision*, pages 740–755, 2014.

- C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [23] Baozhen Liu, Hang Wu, Weihua Su, Wenchang Zhang, and Jinggong Sun. Rotation-invariant object detection using sector-ring hog and boosted random ferns. *The Visual Computer*, 34(5):707–719, 2018.
- [24] Kun Liu, Henrik Skibbe, Thorsten Schmidt, Thomas Blein, Klaus Palme, Thomas Brox, and Olaf Ronneberger. Rotation-invariant hog descriptors using fourier analysis in polar and spherical coordinates. *International Journal of Computer Vision*, 106(3):342–364, 2014.
- [25] Lei Liu, Zongxu Pan, and Bin Lei. Learning a rotation invariant detector with rotatable bounding box. *arXiv preprint arXiv:1711.09405*, 2017.
- [26] Yang Liu, Lei Huang, Xianglong Liu, and Bo Lang. A novel rotation adaptive object detection method based on pair hough model. *Neurocomputing*, 194:246–259, 2016.
- [27] Rodrigo Lozuwa. lozuwa/imp. <https://github.com/lozuwa/imp>, Mar 2019.
- [28] Daniel Mas Montserrat, Qian Lin, Jan Allebach, and Edward J Delp. Training object detection and recognition cnn models using data augmentation. *Electronic Imaging*, 2017(10):27–36, 2017.
- [29] Open-Mmlab. Why does the bounding box become so loose after rotation data aug? is there a better way · issue 4070 · open-mmlab/mmdetection. <https://github.com/open-mmlab/mmdetection/issues/4070>.
- [30] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.
- [32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [33] Pranjal Saxena. Data augmentation for custom object detection: Yolo. <https://medium.com/predict/data-augmentation-for-custom-object-detection-15674966e0c8>, Sep 2020.
- [34] Uwe Schmidt and Stefan Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2050–2057. IEEE, 2012.
- [35] Dennis Soemers. How data augmentation like rotation affects the quality of detection? <https://ai.stackexchange.com/questions/9935/how-data-augmentation-like-rotation-affects-the-quality-of-detection>, Mar 2018.
- [36] Jacob Solawetz. Why and how to implement random rotate data augmentation. <https://blog.roboflow.com/why-and-how-to-implement-random-rotate-data-augmentation/>, Aug 2020.
- [37] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.
- [38] Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1542–1547. IEEE, 2018.
- [39] Matlab Team. Augment bounding boxes for object detection. <https://www.mathworks.com/help/deeplearning/ug/bounding-box-augmentation-using-computer-vision-toolbox.html>.
- [40] Aleksei Tiulpin. Solt: Streaming over lightweight transformations, July 2019.
- [41] Guoli Wang, Bin Fan, and Chunhong Pan. Ordinal pyramid pooling for rotation invariant object recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1349–1353. IEEE, 2015.
- [42] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.
- [43] Yue Xi, Jiangbin Zheng, Xiuxiu Li, Xinying Xu, Jinchang Ren, and Gang Xie. Sr-pod: sample rotation based on principal-axis orientation distribution for data augmentation in deep object detection. *Cognitive Systems Research*, 52:144–154, 2018.
- [44] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.
- [45] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data

augmentation strategies for object detection. *arXiv preprint arXiv:1906.11172*, 2019.