

# On the Robustness of Object Detection Models in Aerial Images

Haodong He<sup>1</sup> Jian Ding<sup>1</sup> Gui-Song Xia<sup>1,2 \*</sup>

<sup>1</sup>NERCMS, School of Computer Science, Wuhan University, China

<sup>2</sup>State Key Lab. of LIESMARS, Wuhan University, China

{haodonghe, jian.ding, guisong.xia}@whu.edu.cn

## Abstract

The robustness of object detection models is a major concern when applied to real-world scenarios. However, the performance of most object detection models degrades when applied to images subjected to corruptions, since they are usually trained and evaluated on clean datasets. Enhancing the robustness of object detection models is of utmost importance, especially for those designed for aerial images, which feature complex backgrounds, substantial variations in scales and orientations of objects. This paper addresses the challenge of assessing the robustness of object detection models in aerial images, with a specific emphasis on scenarios where images are affected by clouds. In this study, we introduce two novel benchmarks based on DOTA-v1.0. The first benchmark encompasses 19 prevalent corruptions, while the second focuses on cloud-corrupted images—a phenomenon uncommon in natural pictures yet frequent in aerial photography. We systematically evaluate the robustness of mainstream object detection models and perform numerous ablation experiments. Through our investigations, we find that enhanced model architectures, larger networks, well-crafted modules, and judicious data augmentation strategies collectively enhance the robustness of aerial object detection models. The benchmarks we propose and our comprehensive experimental analyses can facilitate research on robust object detection in aerial images. Codes and datasets are available at: (<https://github.com/hehaodong530/DOTA-C>)

## 1. Introduction

In recent years, Deep Convolutional Neural Networks (DCNNs) have reached the state-of-the-art level in the field of object detection [14, 19, 13, 41, 42, 40, 31], and gradually replaced the application of traditional methods [49, 50, 5, 9, 10, 11]. Most of the previous research focused on the conditions of independent and identically dis-

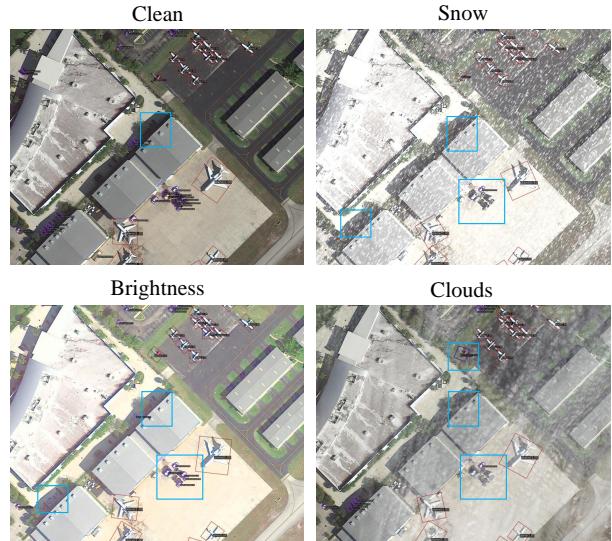


Figure 1. The outcomes of ROI Transformer [7] on a clean image and the same image subject to various corruptions, including Snow and Brightness with a severity of 3, as well as Clouds. We crop the images to make visualization better and the blue boxes contain objects that are detected incorrectly or missed. It should be noted that these images only show the bounding boxes whose confidence scores are not less than 0.3 and the model was trained on clean data without exposure to these corruptions during training. Due to the presence of corruptions in the image, there is a notable increase in false detection and missed detection.

tributed (IID) data, training and evaluating models on clean and high-quality images. However, in real-world scenarios, the quality of images is often affected by weather conditions, the camera itself, and other factors, which means that out-of-distribution (OOD) data are always encountered. Furthermore, the performance of models often suffers a lot from image corruptions [8, 56, 21], as evidenced by the illustrative observations presented in Figure 1.

To evaluate the robustness of deep learning models, several robustness benchmarks in natural images have been

\*Corresponding author:guisong.xia@whu.edu.cn.

proposed. Hendrycks *et al.* [21] established ImageNet-C, a benchmark that expands corruption robustness for object detection. Michaelis [36] *et al.* used these methods to build Pascal-C, COCO-C, and Cityscapes-C, evaluating a few models in autonomous driving. Apart from the benchmarks, there are also several studies on the robustness of object detection models in natural images. In [2, 39], Aharon *et al.* and Benjamin *et al.* proved that Convolutional Neural Networks (CNNs) are less robust than the human vision system. Experiments also indicated that CNNs often perform poorly on novel corruption types despite being trained on different kinds of corruptions [12].

In aerial object detection, there is a lack of benchmarks to evaluate models’ robustness. The publicly available datasets, such as UCAS-AOD [58], HRSC2016 [34], and DOTA [51], are all selected and post-possessed. Therefore, we build a benchmark by utilizing the image transformations presented by Hendrycks *et al.* [21]. The benchmark is based on DOTA-v1.0 [51] and corrupted with 19 corruptions, each spanning five levels of severity.

Furthermore, it is worth noting that clouds are very common in aerial images but rarely appear in natural images. Based on DOTA-v1.0 [51], we introduce a new corruption type called "Clouds" for the second benchmark. Given the dissimilarities between synthetically generated clouds and actual atmospheric conditions, we opt to perform cloud transference from authentic satellite images to clean images for more faithful representation.

To summarize, we present two benchmarks in this work. The first consists of 19 common corruptions, while the second dataset is focused on a singular corruption category, designated as "Clouds." Within the scope of these benchmarks, we undertake a comprehensive evaluation of the robustness exhibited by several established object detection models. More specifically, we aim to determine how well these models can perform on unseen corruptions, so the corruptions mentioned above will not appear as a part of data augmentation strategies during the models’ training phase. Moreover, we engage in an array of ablation experiments, encompassing factors such as alterations in backbone architecture, variations in the number of parameters within the same backbone, and the application of diverse data augmentation strategies, among others. Our contributions are as follows:

- We build two robustness benchmarks for object detection in aerial images based on DOTA-v1.0. The first consists of 19 common corruptions, and the second consists of a cloud noise specifically designed for aerial images.
- We evaluate the robustness of numerous object detection models on the proposed benchmarks and find that

their performances are severely decreased on the out-of-distribution data.

- We conduct numerous analyses and ablation studies on the current models on the robustness benchmark and found that larger model capacity, rotation-invariant modeling, better model architecture designs, and data augmentation strategies can improve the robustness of models. These analyses can provide insights for future studies of robust object detectors in aerial images.

## 2. Related Work

**Aerial Object Detection.** The task of object detection is to predict the bounding box coordinates of objects and the categories in given images. Compared to detection in natural images, aerial object detection is more challenging because of the bird’s eye view, the highly complex backgrounds, the large variations in object scale, and the arbitrary orientations. Since the models used in this work are all based on deep learning, we only review some relevant works focusing on deep neural networks instead of algorithms based on manual features. In [43], Ševo *et al.* applied a convolutional neural network based method to automatic detection in aerial images. Sommer *et al.* [44] investigated the accuracy of Fast R-CNN [13] and Faster R-CNN [41] on aerial image datasets. To solve the problem of arbitrary orientations of objects in aerial images, some methods use Rotated RoI Align [7, 35] to extract accurate region features aligned with the orientation. RRPN [35] designed *rotated anchors* to generate rotated proposals for Rotated RoI Align, which is computationally expensive. RoI Transformer [7] was then proposed to get rotated proposals efficiently. However, rotated RoI Align can not extract the real rotation-invariant region features since the CNN is not rotation-equivariant. ReDet [17] then uses rotation-equivariant CNN and Rotation-invariant RoI Align to extract real rotation-invariant region features. We will show in the experiments that the rotation invariant features are important for the robustness of object detectors for aerial images. Some subsequent works for aerial object detection also made remarkable contributions in improving the accuracy of models and reducing the detectors’ number of parameters [54, 16, 52, 22, 26].

**Robustness Against Corruptions.** Corruptions can be broadly divided into adversarial perturbations [25, 38, 1] and common corruptions [8, 12, 2, 21]. In this paper, we will not introduce adversarial perturbations in detail because we mainly study the impact of common corruptions on models’ performance. In [8], Dodge *et al.* proved that CNNs are susceptible to quality distortions, particularly to blur and noise. Geirhos *et al.* [12] confirmed human vision system is more robust than CNNs on object recogni-

nition under several image degradations. The research of Azulay *et al.* [2] showed that CNNs' accuracy would drop dramatically because of slight geometric transformations. In [21], Hendrycks *et al.* introduced 19 common corruptions and applied them to the ImageNet dataset [6]. Their findings indicated that while ResNet [20] performs better than AlexNet [24] in detecting objects on corrupted images, the models' robustness is nearly identical. In this work, we adopt all corruptions delineated within [21] to DOTA-v1.0 [51] and propose a corruption type called "Clouds". We also investigate the robustness of several object detection models on the corrupted data.

**Corruption Robustness Improvement.** There are several methods to improve models' accuracy on corrupted images. One of them is to remove the corruptions from images. He *et al.* [18] used the dark channel prior to remove haze from images and got good results. In [28], SwinIR was proposed to restore high-quality images from downsampled, noisy, compressed images and performed better than state-of-the-art methods in the case of reduced parameter quantity. However, both traditional methods [45, 4, 18] and deep learning-based methods [27, 55, 23, 28] can only handle one or a few image degradations. These approaches can't generalize to other types of distortions. Another method is to add corrupted data into models' training set. Vasiljevic *et al.* [48] observed that CNNs can enhance their performance in object detection for data affected by the same corruption, by undergoing fine-tuning on blurred images. Geirhos *et al.* [12] revealed that models, when trained on specific corruption types, could outperform the human vision system on those exact corruption types. However, these models exhibited notably limited generalization capabilities when evaluated on different corruption types. In [21], Hendrycks *et al.* reported several methods to improve models' robustness: Histogram Equalization, Multiscale Networks, Larger Networks and so on. In this work, we conduct a series of ablation experiments to figure out how to improve the robustness of models.

## 3. Experiments

### 3.1. Experimental Setup

**Corruptions from ImageNet-C.** We use all the corruptions from ImageNet-C dataset [21] which contains 19 types of corruptions and each one has 5 levels to assess its severity. As defined by Hendrycks *et al.* [21], all corruptions can be divided into four categories: Noise: Gaussian, Shot, Impulse and Speckle; Blur: Defocus, Glass, Motion, Zoom and Gaussian; Weather: Snow, Frost, Fog, Brightness and Spatter; Digital: Contrast, Elastic transform, Pixelate, JPEG compression and Saturate. See Figure 2 for an illustration.

**Clouds.** Since the existing cloud images generated based on GAN [15] and its derivative models can not simulate the state of clouds in nature well, we choose to use "Cloudy Image Arithmetic" [53] which can transfer the clouds to clean images from cloudy images. This process can be divided into two parts: "Cloud Self-Subtraction" and "Cloud Addition-to-Scene". The former part is to extract clouds from a cloudy image:

$$I_{dc}(m, n) = \text{ReLU}[(I_{cs} - \Gamma I)(m, n)] \quad (1)$$

In this context,  $I_{cs}$  denotes a cloudy image of size  $M \times N$  that covers both cloud-free and cloudy sub-areas.  $\Gamma$  presents an  $M \times N$  matrix with all one elements. A pixel with intensity less than a threshold value  $\Gamma$  is categorized into background. The purpose of the  $\text{ReLU}$  function is to make all values greater than 0 unchanged and set all values less than 0 to 0.  $I_{dc}$  is a degraded cloud representation. The process in 1 degrades the both the background pixels and cloud pixels. Therefore, to get cloud ingredient image ( $I_{ci}$ ), we need to give it a compensation coefficient:

$$I_{ci} = \frac{\sum_{m=1}^M \sum_{n=1}^N I_{cs}(m, n) \text{bool}[I_{dc}(m, n) \neq 0]}{\sum_{m=1}^M \sum_{n=1}^N I_{dc}(m, n)} I_{dc} \quad (2)$$

In 2,  $\text{bool}$  function is set to 1 if its variable is true, and 0 otherwise. The latter part is to synthesize the cloudy scene image by adding clouds to a clean image and it can be formulated as follows:

$$I_{scs} = I_{cfl}(B I - I_{ci}) + A I_{ci} \quad (3)$$

In 3,  $I_{scs}$  refers to the synthesized cloudy scene image and  $I_{cfl}$  refers to the cloud-free land image.  $A$  means atmospheric light [46], and it is set to 0.99.  $B$  presents maximum gray-level value, 255 here.

The whole process is shown in Figure 3. We modified the method to batch processing the gray and RGB images of different sizes.

**Object detection models.** We assess a few detectors on the benchmarks introduced in this study, and our evaluation is based on the MMRotate toolbox [57]. Among them, Two-Stage models include Rotated Faster R-CNN [41], RoI Transformer [7], Oriented R-CNN [52] and ReDet [17]. One-Stage models include Rotated RetinaNet [30], Rotated FCOS [47], R<sup>3</sup>Det [54] and S<sup>2</sup>A-Net [16]. All these models' backbone is ResNet50 [20] (1024,1024,200) with Feature Pyramid Networks (FPN) [29], except for ReDet, which employs ReResNet50 (1024,1024,200) with ReFPN [17]. Every model has been trained with batch size 8 (2 per GPU), standard hyperparameters, the same image normalization configurations and a schedule for 12 epochs. It is crucial to highlight that these corruptions are exclusively employed on the test set.

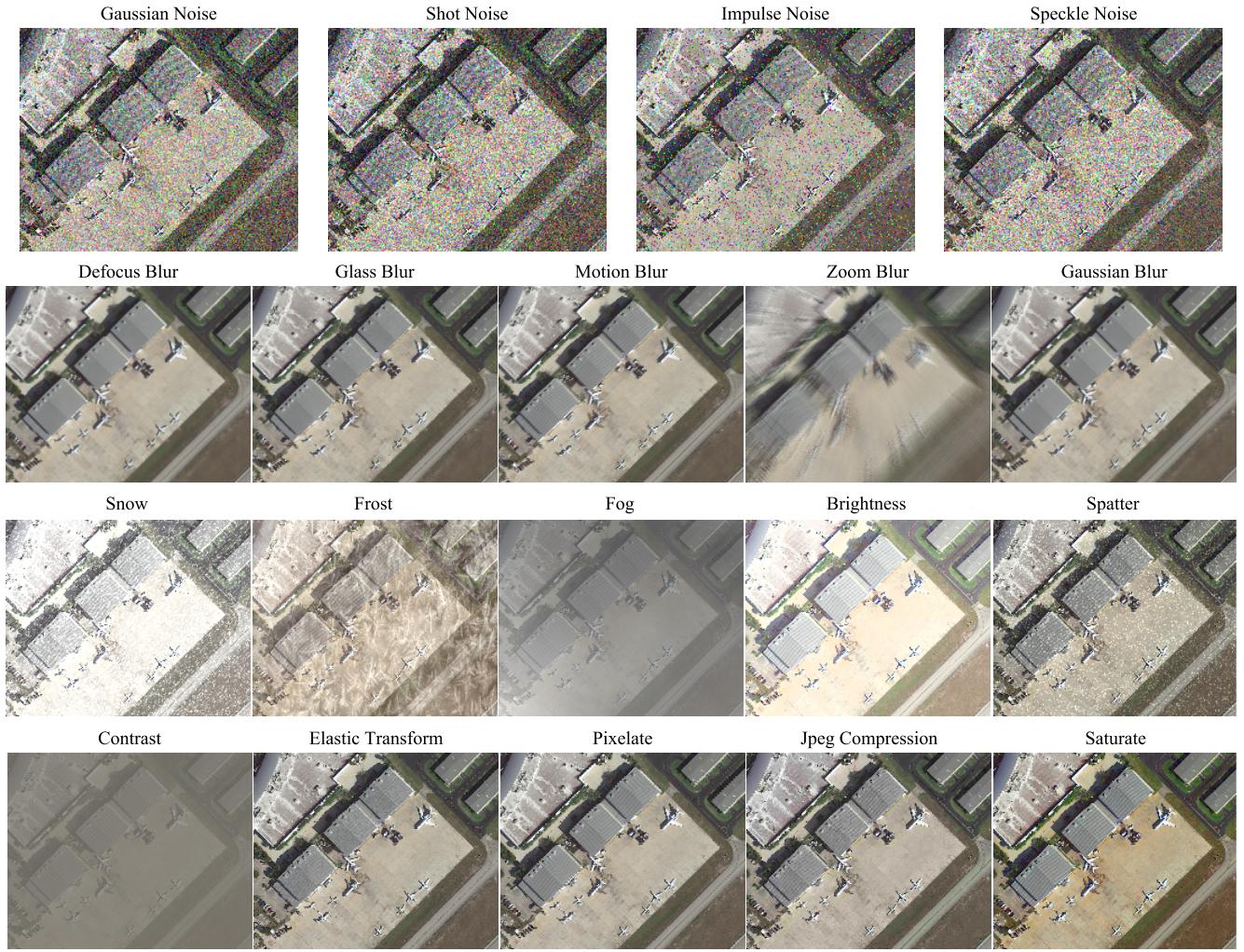


Figure 2. The original image is randomly selected from DOTA-v1.0 [51] test set and corrupted with 19 corruption types, severity 3.

**Datasets.** DOTA-v1.0 [51] is an aerial image dataset, which has 1411 train images, 458 validation images and 937 test images. There are 15 classes of objects labeled in this dataset, and they are plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and basketball court. Following the previous works [7, 17], we use the training set and the validation set for training and the testing set for testing.

### 3.2. Evaluation Metrics

We apply AP<sub>50</sub> as an evaluation metric, which stands for the PASCAL Average Precision metric at 50% Intersection over Union (IoU). Our evaluation of models is based on the mean performance under corruption (mPC) [37], which is formulated as follows:

$$mPC = \frac{1}{N_c} \sum_{c=1}^{N_c} \frac{1}{N_s} \sum_{s=1}^{N_s} AP_{50}^{c,s} \quad (4)$$

Here, the metric AP<sub>50</sub><sup>c,s</sup> represents the performance of a model, evaluated using test data that has been corrupted with corruption type c under severity level s. The values N<sub>c</sub> = 19 and N<sub>s</sub> = 5 denote the number of corruption types and severity levels, respectively. It is important to note that a higher mPC does not inherently signify superior robustness of the model, since its performance might undergo a swifter deterioration in the presence of corruptions compared to a model with a lower mPC. Consequently, to quantify the performance degradation induced by corruptions, relative performance under corruption (rPC) [37] is put forward and defined as below:

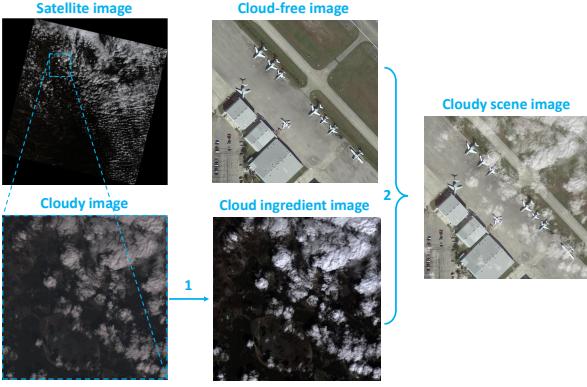


Figure 3. We get satellite images from USGS Landsat 8 Collection 2 Tier 1 Raw Scenes (Landsat-8 image courtesy of the U.S. Geological Survey). The satellite image’s product ID in this figure is “LC08\_119029\_20140908” and we select three bands (“B4”, “B3”, “B2”) to make it a true color image. The size of the cloudy image is 1024x1024, which is cut from the satellite image. Process 1 represents “Cloud Self-Subtraction” and process 2 represents “Cloud Addition-to-Scene”.

$$rPC = \frac{mPC}{AP_{50}^{\text{clean}}} \quad (5)$$

In 5,  $AP_{50}^{\text{clean}}$  means a model’s  $AP_{50}$  on clean test set. Since the clouds are transferred from satellite images to clean images, we don’t grade the corruption severity levels as in ImageNet-C, so the evaluation metrics are only  $AP_{50}^{\text{clouds}}$  and  $rPC_{\text{clouds}}$ . They are defined as follows:

$$rPC_{\text{clouds}} = \frac{AP_{50}^{\text{clouds}}}{AP_{50}^{\text{clean}}} \quad (6)$$

$AP_{50}^{\text{clouds}}$  means a model’s  $AP_{50}$  on data corrupted with clouds and  $rPC_{\text{clouds}}$  presents the relative performance degradation of a model caused by clouds.

### 3.3. Results

**Corruptions from ImageNet-C.** For corruptions from ImageNet-C [21], the models’  $AP_{50}^{\text{clean}}$ , mPC and  $AP_{50}$  for each corruption type averaged over all severity levels are displayed in Table 1. The respective results of models for individual severity levels are shown in supplementary material. Furthermore, we conduct an evaluation and analysis of the degradation in models’ performance on images corrupted with various types of corruptions. The evaluation result rPC can be seen in Figure 4.

The evaluation of the models’ performance with respect to mPC reveals that, ReDet [17] outperforms other Two-Stage models and S<sup>2</sup>A-Net [16] exhibits superior performance among One-Stage models on both clean and cor-

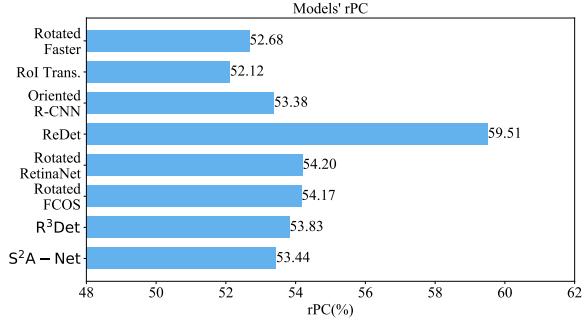


Figure 4. Models’ rPC on DOTA-v1.0 [51] test set corrupted with 19 corruption types from ImageNet-C [21]. The models from top to bottom are Rotated Faster R-CNN [41], RoI Transformer [7], Oriented R-CNN [52], ReDet [17], Rotated RetinaNet [30], Rotated FCOS [47], R<sup>3</sup>Det [54], and S<sup>2</sup>A-Net [16].

rupted test set of DOTA-v1.0. In general, models that perform well on clean data also perform better on corrupted data. However, this trend is not universal, as can be seen for RoI Transformer [7] and Oriented R-CNN [52], where the former outperforms the latter on clean data, but the opposite is true on corrupted data. The brightness corruption has a minimal impact on models’ performance, whereas the zoom blur corruption degrades it the most.

For rPC, ReDet [17] achieves the largest rPC in Two-Stage models and even among all models, while Rotated RetinaNet [30] performs best in One-Stage models. Among them, the performance of RoI Transformer [7] degrades the most rapidly. On the whole, One-Stage models often perform better than Two-Stage models on rPC (except Re-Det). Moreover, we conduct a comprehensive investigation into the influence of 4 different corruption categories (noise corruptions, blur corruptions, weather corruptions, and digital corruptions) on the models’ performance. As shown in Table 2, it is evident that weather corruptions and digital corruptions exhibit minimal impact on the models, while noise corruptions result in the poorest performance of them. Among the considered models, Redet exhibits the most gradual performance degradation when subjected to any of the four types of corruptions. The other models show marginal differences in rPCnoise, rPCblur, rPCweather, and rPCdigital.

**Clouds.** The performance of models can be seen in Table 3. ReDet achieves the largest  $AP_{50}^{\text{clouds}}$  and  $rPC_{\text{clouds}}$  in Two-Stage models, while S<sup>2</sup>A-Net achieves the largest  $AP_{50}^{\text{clouds}}$  and Rotated FCOS achieves the largest  $rPC_{\text{clouds}}$  in One-Stage models.

### 3.4. Ablation Study

Based on RoI Transformer [7], we conduct a series of ablation experiments to evaluate the impact of various fac-

Model	Noise						Blur						Weather						Digital			
	AP <sub>50</sub> <sup>clean</sup>	mPC	Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.	
Rot. Faster [41]	73.4	38.7	20.2	19.7	17.7	27.6	40.5	46.4	36.0	14.1	43.0	24.3	46.2	49.3	63.1	46.7	42.4	33.2	53.1	50.4	60.7	
RoI Trans. [7]	76.1	39.7	19.8	20.2	17.8	29.1	41.1	48.8	37.6	14.7	44.0	26.5	47.1	49.2	63.5	49.4	42.5	35.0	53.6	51.5	62.3	
Ori. R-CNN [52]	75.7	40.4	21.7	21.7	18.7	30.3	41.9	49.0	37.6	14.8	44.3	25.6	48.7	51.5	65.6	48.5	43.2	34.9	55.5	50.7	63.4	
ReDet [17]	76.7	45.6	24.7	24.6	22.4	34.3	50.3	53.6	42.5	18.1	53.2	35.3	58.3	63.1	70.5	52.0	54.3	33.2	59.4	52.4	64.9	
Rot. Retina. [30]	68.4	37.1	20.0	19.7	16.9	26.7	40.5	45.8	34.9	14.0	43.3	23.3	45.2	47.9	59.4	42.9	40.3	31.5	48.0	46.4	58.1	
Rot. FCOS [47]	71.3	38.6	20.6	20.5	18.7	27.6	41.3	46.8	35.2	14.5	43.7	26.1	46.6	50.7	61.2	45.8	43.8	31.6	51.4	48.3	59.6	
R <sup>3</sup> Det [54]	69.8	37.6	19.9	19.6	17.3	27.4	38.6	44.3	33.9	14.4	42.0	24.8	46.5	48.6	61.1	43.8	41.8	31.8	51.7	47.1	59.4	
S <sup>2</sup> A-Net [16]	73.9	39.5	18.6	18.6	15.7	26.3	42.3	48.4	36.0	15.1	44.9	28.7	49.7	53.2	64.0	46.5	45.0	33.8	50.9	49.9	62.7	

Table 1. In this table, models’ AP<sub>50</sub><sup>clean</sup>, mPC and averaged AP<sub>50</sub> for each corruption type over all severity levels on corrupted DOTA-v1.0 [51] test set are displayed. The corruptions from left to right are Noise: Gaussian, Shot, Impulse and Speckle; Blur: Defocus, Glass, Motion, Zoom and Gaussian; Weather: Snow, Frost, Fog, Brightness and Spatter; Digital: Contrast, Elastic transform, Pixelate, JPEG compression and Saturate. The models from top to bottom are Two-Stage models: Rotated Faster R-CNN [41], RoI Transformer [7], Oriented R-CNN [52], and ReDet [17]; One-Stage models: Rotated RetinaNet [30], Rotated FCOS [47], R<sup>3</sup>Det [54], and S<sup>2</sup>A-Net [16].

Model	rPC <sub>noise</sub> (%)	rPC <sub>blur</sub> (%)	rPC <sub>weather</sub> (%)	rPC <sub>digital</sub> (%)
Rot. Faster [41]	29.01	49.06	62.56	65.38
RoI Trans. [7]	28.55	48.94	61.95	64.35
Ori. R-CNN	30.52	49.58	63.39	65.45
ReDet [17]	34.55	56.77	72.81	68.91
Rot. Retina. [30]	30.40	52.18	63.93	65.55
Rot. FCOS [47]	30.67	50.88	64.62	65.85
R <sup>3</sup> Det [54]	30.14	49.64	64.38	66.43
S <sup>2</sup> A-Net [16]	26.83	50.55	65.50	65.57

Table 2. Models’ rPC for noise corruptions, blur corruptions, weather corruptions, and digital corruptions over all severity levels on corrupted DOTA-v1.0 [51] test set.

Model	AP <sub>50</sub> <sup>clouds</sup>	rPC <sub>clouds</sub> (%)
Rotated Faster R-CNN [41]	58.53	79.73
RoI Transformer [7]	60.03	78.90
Oriented R-CNN	60.59	80.05
ReDet [17]	66.19	86.33
Rotated RetinaNet [30]	55.12	80.55
Rotated FCOS [47]	57.51	80.68
R <sup>3</sup> Det [54]	56.65	81.15
S <sup>2</sup> A-Net [16]	59.29	80.22

Table 3. Models’ AP<sub>50</sub><sup>clouds</sup> and rPC<sub>clouds</sub> on DOTA-v1.0 [51] test set corrupted with clouds.

tors on the model’s performance. Specifically, we analyzed the effects of different backbones, varying capacity of the same backbone, rotation-invariant modeling, and data augmentation strategies. The results of all these experiments are presented in Table 4 and the respective results of them for individual severity levels are shown in supplementary material.

**Ablation of Backbones.** We choose ResNet50 [20], Swin-Transformer-Tiny (Swin-T) [32], ConvNeXt-Tiny (ConvNeXt-T) [33] as RoI Transformer’s [7] backbone and test their performance on clean DOTA-v1.0 [51] test set and corrupted one respectively. All models are trained on clean DOTA-v1.0 for 12 epochs. It’s worth emphasizing that all the three backbones are pre-trained on ImageNet-1K [6] and

their parameters can be seen in Table 5.

We evaluate the performance of different RoI Transformer models, as demonstrated in Table 6. Among these models, RoI Transformer (Swin-T) achieves the largest AP<sub>50</sub><sup>clean</sup>, while RoI Transformer (ConvNeXt-T) outperforms the others in terms of other metrics. Notably, RoI Transformer (ConvNeXt-T) degrades the most gracefully in the presence of corruptions, while the robustness of RoI Transformer (ResNet50) is the worst among the three models. However, taking ConvNeXt-T as the backbone fails to enhance the model’s accuracy on clean data, and even leads to a slight decrease. Nevertheless, Table 4 reveals that RoI Transformer (ConvNeXt-T) performs much better than the other two models on data corrupted with Gaussian, shot, impulse and speckle noise. Its performance even matches that of RoI Transformer (Swin-L). Given that RoI Transformer (Swin-T) and RoI Transformer (ConvNeXt-T) share identical resolutions, parameter counts, FLOPs, and classifier heads, it is justifiable to hypothesize that the latter’s architecture exhibits higher stability and resilience against corruptions compared to the former.

**Different backbone capacity.** On the basis of Swin-Transformer [32] as the backbone of RoI Transformer, we conduct a series of experiments to investigate the impact of different backbone capacities on the model’s performance. All models were trained on clean DOTA-v1.0 for 12 epochs. Specifically, we evaluate the performance of RoI Transformer using Swin-Transformer-Tiny (Swin-T), Swin-Transformer-Small (Swin-S), Swin-Transformer-Base (Swin-B), and Swin-Transformer-Large (Swin-L), all of which were pre-trained on ImageNet-1K except for Swin-L (which was pre-trained on ImageNet-22K and then finetuned to ImageNet-1K). The parameters for each of the backbones are provided in Table 5.

The results of the experiments are presented in Table 7. We can see that AP<sub>50</sub><sup>clean</sup> has no clear positive correlation with models’ backbone capacity. However, other metrics are getting better as backbone capacity increases, including

Method	Noise						Blur						Weather						Digital			
	AP <sub>50</sub> <sup>clean</sup>	mPC	Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.	
ResNet50 [20]	76.1	39.7	19.8	20.2	17.8	29.1	41.1	48.8	37.6	14.7	44.0	26.5	47.1	49.2	63.5	49.4	42.5	35.0	53.6	51.5	62.3	
ConvNeXt-T [33]	75.0	47.5	33.3	33.1	31.9	43.2	47.8	52.0	42.6	17.7	50.7	37.0	56.3	63.5	68.9	54.8	56.0	32.0	56.7	58.5	65.8	
Swin-T [32]	77.5	43.1	26.3	25.8	25.9	35.9	45.6	48.7	41.3	15.6	48.4	34.9	53.4	59.4	67.4	55.1	48.3	37.4	53.9	55.7	62.4	
Swin-S [32]	77.1	44.3	26.3	25.8	25.9	35.9	45.6	48.7	41.3	15.6	48.4	34.9	53.4	59.4	67.4	55.1	48.3	37.4	53.9	55.7	62.4	
Swin-B [32]	77.7	44.8	26.6	26.6	27.4	36.4	47.1	50.7	41.0	15.0	49.7	35.5	54.5	58.0	68.7	55.2	48.5	37.6	50.6	57.4	64.5	
Swin-L [32]	77.6	47.5	32.5	32.1	33.9	42.1	46.8	50.1	41.5	16.4	49.9	35.9	58.3	63.8	70.6	57.7	53.1	37.1	55.3	58.6	66.5	
RandomRotate	76.4	40.9	21.5	21.3	18.6	29.5	44.1	50.5	40.6	15.3	46.3	28.3	47.9	52.1	65.0	47.9	43.1	35.4	56.8	50.7	62.9	
Mosaic [3]	74.4	38.8	21.3	20.7	18.1	28.3	41.3	46.7	36.5	13.8	44.2	25.9	46.3	49.7	62.3	46.4	41.8	32.3	51.4	49.0	61.9	

Table 4. This table shows RoI Transformer’s [7] AP<sub>50</sub><sup>clean</sup>, mPC ,and averaged AP<sub>50</sub> for each corruption type over all severity levels on corrupted DOTA-v1.0 [51] test set, after many methods were used to improve it. ResNet50 [20], ConvNeXt-T (ConvNeXt-Tiny) [33], Swin-T (Swin-Transformer-Tiny) [32], Swin-S (Swin-Transformer-Small), Swin-B (Swin-Transformer-Base), and Swin-L (Swin-Transformer-Large) refer to RoI Transformer with different backbones. RandomRotate and Mosaic [3] refer to two data augmentation strategies used by the model, respectively.

name	resolution	#params	FLOPs
ResNet50 [20]	224x224	26M	3.5G
ConvNeXt-T [33]	224x224	28M	4.5G
Swin-T [32]	224x224	28M	4.5G
Swin-S [32]	224x224	50M	8.7G
Swin-B [32]	224x224	88M	15.4G
Swin-L [32]	224x224	197M	34.5G

Table 5. ResNet50 [20], ConvNeXt-T [33], Swin-T [32], Swin-S, Swin-B, and Swin-L’s parameters. FLOPs stands for floating point of operations and it’s a measure of model complexity.

Model	AP <sub>50</sub> <sup>clean</sup>	mPC	rPC(%)	AP <sub>50</sub> <sup>clouds</sup>	rPC <sub>clouds</sub> (%)
ResNet50 [20]	76.08	39.66	52.12	60.03	78.90
Swin-T [32]	77.51	43.13	55.64	62.83	81.06
ConvNeXt-T [33]	74.98	47.47	63.31	64.52	86.04

Table 6. ResNet50 [20], Swin-T [32], and ConvNeXt-T [33] stand for RoI Transformer [7] (ResNet50), RoI Transformer (Swin-T), and RoI Transformer (ConvNeXt-T) respectively.

Model	AP <sub>50</sub> <sup>clean</sup>	mPC	rPC(%)	AP <sub>50</sub> <sup>clouds</sup>	rPC <sub>clouds</sub> (%)
Swin-T [32]	77.51	43.13	55.64	62.83	81.06
Swin-S [32]	77.12	44.28	57.42	63.27	82.05
Swin-B [32]	77.70	44.79	57.64	64.84	83.45
Swin-L [32]	77.65	47.49	61.15	66.73	85.94

Table 7. Swin-T [32], Swin-S, Swin-B, and Swin-L stand for RoI Transformer [7] (Swin-T), RoI Transformer (Swin-S), RoI Transformer (Swin-B), and RoI Transformer (Swin-L) [7] respectively.

mPC, rPC, AP<sub>50</sub><sup>clouds</sup>, and rPC<sub>clouds</sub>. The results indicate that incorporating a larger and deeper backbone into the model contributes positively to enhancing its robustness, under the assumption of keeping other influencing factors unchanged.

**Two-Stage v.s. One-Stage.** The models’ performance across different metrics is presented in Table 1, 3 and Figure 4. ReDet [17] performs the best among all the models mentioned in these tables and figure. Apart from it, the Two-Stage models achieve higher AP<sub>50</sub><sup>clean</sup>, mPC and AP<sub>50</sub><sup>clouds</sup>, while One-Stage models show better performance on rPC

Model	AP <sub>50</sub> <sup>clean</sup>	mPC	rPC(%)	AP <sub>50</sub> <sup>clouds</sup>	rPC <sub>clouds</sub> (%)
Faster R-CNN [41]	73.41	38.67	52.68	58.53	79.73
RoI Transformer [7]	76.08	39.66	52.12	60.03	78.90
ReDet [17]	76.68	45.63	59.51	66.19	86.33

Table 8. Faster R-CNN [41] in this table refers to Rotated Faster R-CNN, and the backbone of the models above is ResNet50 [20].

and rPC<sub>clouds</sub>. The results indicate that Two-Stage models tend to outperform One-Stage models under normal circumstances, but their performance deteriorates more quickly than that of the latter under the influence of corruptions.

**Ablation of modules.** Based on the rotation-equivariant features, Rotation-invariant RoI Align (RiRoI Align) is used in ReDet [17]. This module can adaptively extract rotation-invariant features from equivariant features according to the orientation of the region of interest (RoI), but Rotated Faster R-CNN [41] and RoI Transformer [7] do not integrate it. Their performance is show in Table 8. ReDet preforms a little better than the other two models in AP<sub>50</sub><sup>clean</sup> and performs much better in mPC, rPC, AP<sub>50</sub><sup>clouds</sup> and rPC<sub>clouds</sub>. This observation suggests that using rotation-invariant modeling can enhance the robustness of object detection models.

**Ablation of data augmentations.** Our aim is to explore the effects of data augmentations unrelated to these corruptions on models’ robustness. In order to ensure that these corruptions are unseen to the models during training, we choose two data augmentation strategies, RandomRotate and Mosaic [3]. RandomRotate means to rotate the images randomly and in Mosaic, four training images are combined in a certain scale to form a single image. Figure 5 shows the effect after they are applied to an image and models’ performance is displayed in Table 9. The results suggest an improvement in models’ robustness with the adoption of the two data augmentation strategies. Compared with Mosaic, using RandomRotate is a better choice, because it has better performance on clean data and it is more robust against corrupted images than the model using Mosaic.

Augmentation	$AP_{50}^{\text{clean}}$	mPC	rPC(%)	$AP_{50}^{\text{clouds}}$	rPC <sub>clouds</sub> (%)
RoI Transformer [7]	76.08	39.66	52.12	60.03	78.90
RandomRotate	76.38	40.93	53.59	61.26	80.21
Mosaic [3]	74.36	38.84	52.23	59.56	80.10

Table 9. RandomRotate and Mosaic [3] refer to RoI Transformer [7] (RandomRotate) and RoI Transformer (Mosaic), respectively.

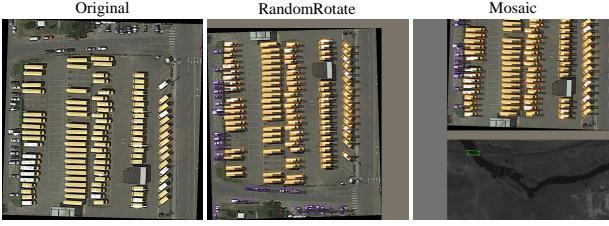


Figure 5. From left to right, it is the original image, the image after using RandomRotate and the image after using Mosaic.

**Discussion.** In general, the effectiveness of the methods mentioned above in improving the performance of RoI Transformer [7] on  $AP_{50}^{\text{clouds}}$  is not guaranteed and may vary.

In addition, Figure 6 reveals that all the proposed methods are effective in enhancing the robustness of RoI Transformer. Among them, replacing the backbone with ConvNeXt-T or Swin-L and RoI Transformer (ResNet50) using RiRoI Align module exhibit the most significant improvement, whereas the utilization of data augmentation exhibits minimal impact. With the increase of parameters (from Swin-T to Swin-L), the models' rPC and rPC<sub>clouds</sub> become better and better on the whole.

We also conduct an evaluation and analysis of how different corruption categories (noise corruptions, blur corruptions, weather corruptions, and digital corruptions) affect the performance of the models which use the proposed methods and the results are shown in Table 10. It is apparent that RoI Transformer (ConvNeXt-T [33]) exhibits the most promising performance on rPC<sub>noise</sub>, rPC<sub>weather</sub>, and rPC<sub>digital</sub>, while ReDet [17] achieves the highest rPC<sub>blur</sub>. That means changing the backbone architecture proves to be the most effective method in improving RoI Transformer's performance on noise, weather, and digital corruptions, while using RiRoI Align module is the most helpful in improving the accuracy of it on data corrupted with blur corruptions. Moreover, as the backbone's parameters increase (from Swin-T to Swin-L), the robustness of the model against the 4 corruptions shows a gradual improvement, but there are exceptions, such as rPC<sub>weather</sub> and rPC<sub>digital</sub> of RoI Transformer (Swin-B) are lower than those of RoI Transformer (Swin-S). Incidentally, the influence of data augmentations on models' performance in the presence of these corruptions is constrained and the model using Mosaic even performs worse on rPC<sub>digital</sub>.

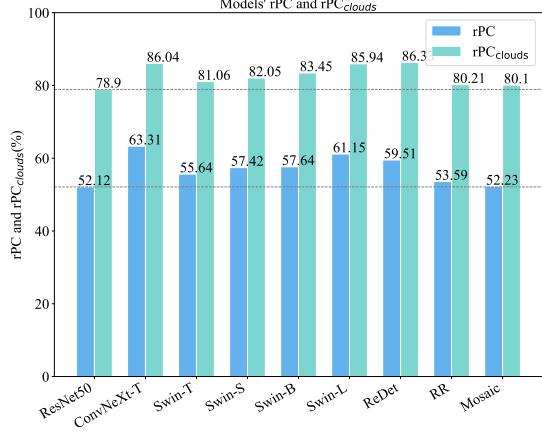


Figure 6. After applying these methods to RoI Transformer [7], the rPC and rPC<sub>clouds</sub> values of the models are as shown in the figure. The backbones of the models using RR (RandomRotate) and Mosaic are both ResNet50 [20]. The gray lines in the figure indicate that rPC and rPC<sub>clouds</sub> of RoI Transformer (ResNet50) are used as baselines.

Method	rPC <sub>noise</sub> (%)	rPC <sub>blur</sub> (%)	rPC <sub>weather</sub> (%)	rPC <sub>digital</sub> (%)
ResNet50 [20]	28.55	48.94	61.95	64.35
ConvNeXt-T [33]	47.17	56.26	74.82	71.77
Swin-T [32]	34.86	50.90	67.95	64.73
Swin-S [32]	36.93	51.75	70.06	66.83
Swin-B [32]	37.66	52.41	69.99	66.53
Swin-L [32]	45.28	52.71	73.74	69.71
ReDet [17]	34.55	56.77	72.81	68.91
RandomRotate	29.75	51.53	63.13	65.16
Mosaic [3]	29.71	49.07	62.05	63.60

Table 10. After applying these methods to RoI Transformer [7], models' rPC for noise corruptions, blur corruptions, weather corruptions, and digital corruptions over all severity levels on corrupted DOTA-v1.0 [51] test set.

## 4. Conclusion

In this paper, we propose two new benchmarks for investigating the robustness of models on corrupted data in aerial object detection. We apply all the image transformations mentioned in ImageNet-C dataset, as well as clouds that are often present in aerial images but rare in natural images, to generate corruptions. Then, we conduct a detailed evaluation of several mainstream models' robustness on these corrupted images, and find that the models suffer varying degrees of performance degradation when detecting objects on these images. Finally, a series of ablation experiments are carried out to demonstrate that some modifications can effectively improve the robustness of a model, such as better model structure, larger networks, more appropriate modules and data augmentation strategies. We hope that our benchmark and study can provide some help for future research on the the robustness of models in aerial object detection.

## References

- [1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *CVPR*, pages 3389–3398, 2018. 2
- [2] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv:1805.12177*, 2018. 2, 3
- [3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020. 7, 8, 13, 15, 16
- [4] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR 2005*, volume 2, pages 60–65. Ieee, 2005. 3
- [5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*, volume 1, pages 886–893. Ieee, 2005. 1
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, pages 248–255. Ieee, 2009. 3, 6
- [7] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *CVPR*, pages 2849–2858, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14
- [8] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *QoMEX 2016*, pages 1–6. IEEE, 2016. 1, 2
- [9] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR 2008*, pages 1–8. Ieee, 2008. 1
- [10] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *CVPR 2010*, pages 2241–2248. Ieee, 2010. 1
- [11] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009. 1
- [12] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *NeurIPS*, 31, 2018. 2, 3
- [13] Ross Girshick. Fast r-cnn. In *CVPR*, pages 1440–1448, 2015. 1, 2
- [14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *TPAMI*, 38(1):142–158, 2015. 1
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020. 3
- [16] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *TGRS*, 2021. 2, 3, 5, 6, 14
- [17] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *CVPR*, pages 2786–2795, 2021. 2, 3, 4, 5, 6, 7, 8, 14, 16
- [18] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *TPAMI*, 33(12):2341–2353, 2010. 3
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 1
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 6, 7, 8, 12, 15, 16
- [21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv:1903.12261*, 2019. 1, 2, 3, 5, 11, 12
- [22] Liping Hou, Ke Lu, Jian Xue, and Yuqiu Li. Shape-adaptive selection and measurement for oriented object detection. In *CVPR*, 2022. 2
- [23] Filippos Kokkinos and Stamatios Lefkimiatis. Iterative joint image demosaicking and denoising using a residual denoising network. *TIP*, 28(8):4177–4188, 2019. 3
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. 3
- [25] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv:1611.01236*, 2016. 2
- [26] Wentong Li, Yijie Chen, Kaixuan Hu, and Jianke Zhu. Oriented repoints for aerial object detection. In *CVPR*, pages 1829–1838, 2022. 2
- [27] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. In *ECCV*, pages 272–289, 2018. 3
- [28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, pages 1833–1844, 2021. 3
- [29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3
- [30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017. 3, 5, 6, 14
- [31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. 1
- [32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6, 7, 8, 12, 13, 15, 16
- [33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 6, 7, 8, 12, 13, 15, 16
- [34] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *ICPRAM*, pages 324–331, 2017. 2

- [35] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *TMM*, 20(11):3111–3122, 2018. 2
- [36] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv:1907.07484*, 2019. 2
- [37] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv:1907.07484*, 2019. 4
- [38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, pages 1765–1773, 2017. 2
- [39] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv:1806.00451*, 2018. 2
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 1, 2, 3, 5, 6, 7, 14
- [42] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1
- [43] Igor Ševo and Aleksej Avramović. Convolutional neural network based automatic object detection on aerial images. *GRSL*, 13(5):740–744, 2016. 2
- [44] Lars Wilko Sommer, Tobias Schuchert, and Jürgen Beyerer. Fast deep vehicle detection in aerial images. In *WACV 2017*, pages 311–319. IEEE, 2017. 2
- [45] Hossein Talebi and Peyman Milanfar. Global image denoising. *TIP*, 23(2):755–768, 2013. 3
- [46] Robby T Tan. Visibility in bad weather from a single image. In *CVPR 2008*, pages 1–8. IEEE, 2008. 3
- [47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 3, 5, 6, 14
- [48] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv:1611.05760*, 2016. 3
- [49] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR 2001*, volume 1, pages I–I. Ieee, 2001. 1
- [50] Paul Viola and Michael J Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004. 1
- [51] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Beßongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018. 2, 3, 4, 5, 6, 7, 8
- [52] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *ICCV*, pages 3520–3529, 2021. 2, 3, 5, 6, 14
- [53] Zunxiao Xu, Kang Wu, Lei Huang, Qimao Wang, and Peng Ren. Cloudy image arithmetic: A cloudy scene synthesis paradigm with an application to deep-learning-based thin cloud removal. *TGRS*, 60:1–16, 2021. 3, 11
- [54] Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3det: Refined single-stage detector with feature refinement for rotating object. In *AAAI*, volume 35, pages 3163–3171, 2021. 2, 3, 5, 6, 14
- [55] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271, 2018. 3
- [56] Yiren Zhou, Sibo Song, and Ngai-Man Cheung. On classification of distorted images with deep convolutional neural networks. In *ICASSP 2017*, pages 1213–1217. IEEE, 2017. 1
- [57] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, Wenwei Zhang, and Kai Chen. Mmrotate: A rotated object detection benchmark using pytorch. In *ACM MM*, 2022. 3, 12
- [58] Haigang Zhu, Xiaogang Chen, Weiqun Dai, Kun Fu, Qixiang Ye, and Jianbin Jiao. Orientation robust object detection in aerial images using deep convolutional neural network. In *ICIP 2015*, pages 3735–3739. IEEE, 2015. 2

We furnish additional details concerning the employed image corruptions and the conducted experiments. To be more specific, we initially showcase images afflicted by a prevalent corruption type across five distinct severity levels as delineated in ImageNet-C [21]. Subsequently, we expound upon the process of introducing cloud corruptions to the images. Moreover, we provide supplementary material that outlines the experimental setup and elucidates the models' performance on corrupted data across various severity levels. Additionally, we report some models' rPC and rPC<sub>clouds</sub> for each object category and do further research on the performance of the methods in ablation experiments.

## A. Corruptions

### A.1. Example of Corruptions From ImageNet-C

In Figure 7, we show the image corrupted with "snow" of its 5 severity levels. From 1 to 5, the objects in the image gradually change from clear to unrecognizable.



Figure 7. Presented in this figure are a clean image and the same image corrupted with "snow" of level 1, 2, 3, 4, and 5.

### A.2. Corrupting Images With Clouds

We acquire 10 satellite images from USGS Landsat 8 Collection 2 Tier 1 Raw Scenes (Landsat-8 image courtesy of the U.S. Geological Survey) and select the three bands "B4", "B3" and "B2" to make them true color images. Subsequently, we crop these images and get 30 cloudy scene images with the size of 1024x1024. There are some examples in Figure 8. This set of cloud-affected images is employed to introduce corruption to the DOTA-v1.0 test set.

In the "Cloudy Image Arithmetic" proposed by Xu *et al.* [53], the size of images is hard-coded and the code can

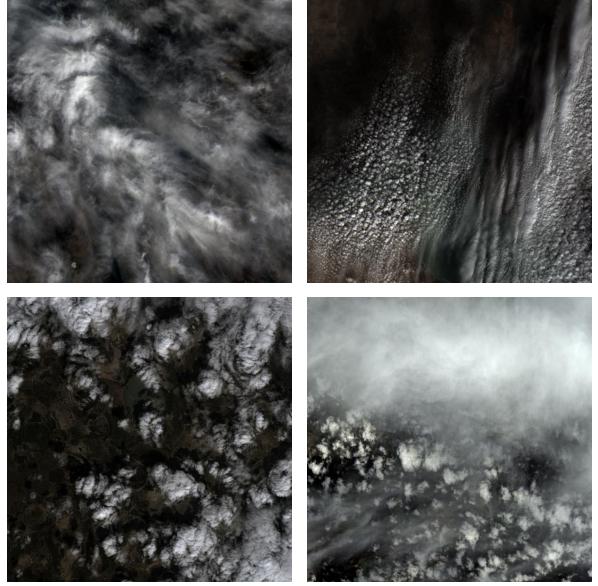


Figure 8. Presented in this figure are different kinds of cloudy scene images.

only deal with the RGB images. We have modified the code to resolve these constraints and now the corruption can be applied to grayscale images and images with varying sizes, which is a necessary prerequisite for corrupting DOTA-v1.0 with clouds. Additionally, batch processing functionality was also added to the code.

## B. Experimental Details

### B.1. Experimental Setup

We train all models on a machine with 2 NVIDIA GeForce RTX 2080 Ti (11019MiB) and 2 NVIDIA TITAN Xp (12196MiB) or a machine with 4 TITAN RTX (24220MiB).

We split the original images to 1024x1024 and set the value of gaps to 200. Preceding the commencement of the training procedure, a series of preprocessing steps are administered to the images. These include resizing, random horizontal flipping, and normalization, after which the images are transformed into RGB format. The training phase spans 12 epochs for all models, adhering to a consistent learning rate policy. This policy encompasses a linear warmup stage, with the learning rate diminishing to one-tenth of its original value after the eighth and eleventh epochs. During the testing phase, the test images undergo a two-step process: initial corruption and subsequent segmentation. Following corruption, images are cropped using the same strategy as before. Subsequently, resizing and normalization are carried out, culminating in the conversion of

the images to RGB format. The entirety of the training and testing procedures are executed through the utilization of the MMRotate toolbox [57].

## B.2. Models’ Performance

The performance of Two-Stage models and One-Stage models on corrupted data of 5 different severity levels can be seen in Figure 11, 12, 13, 14, and 15. And the performance of RoI Transformer using different methods in ablation experiments can be seen in Figure 16, 17, 18, 19, and 20. The visualization of models’ performance on data corrupted with Snow at severity level 3 can be seen in Figure 9 and 10.

## C. Additional Results

We have made a further supplement to the ablation experiments and studied the impact of the methods on the performance of RoI Transformer [7] for each object category.

Figure 21 displays the models’ robustness on corruptions from ImageNet-C [21] for each object category, while Figure 22 displays the models’ robustness on data corrupted with clouds. The results indicate that the models’ robustness is superior in detecting plane and tennis court, while the performance degrades more rapidly for baseball diamond and helicopter. Changing backbone from ResNet50 [20] to ConvNeXt-T [33] has the greatest enhancement on the robustness of the model, while using Mosaic data augmentation strategy only slightly enhances its performance. Furthermore, as the backbone’s parameters increase from Swin-T [32] to Swin-L, the overall rPC is increasing, but it is not the case for a specific object category. For example, the robustness of RoI Transformer (Swin-T) on swimming pool is even better than RoI Transformer (Swin-L). Finally, Comparing RoI Transformer (ResNet50) with Re-Det, we find that rotation-invariant modeling can improve the robustness on almost all object categories except helicopter.

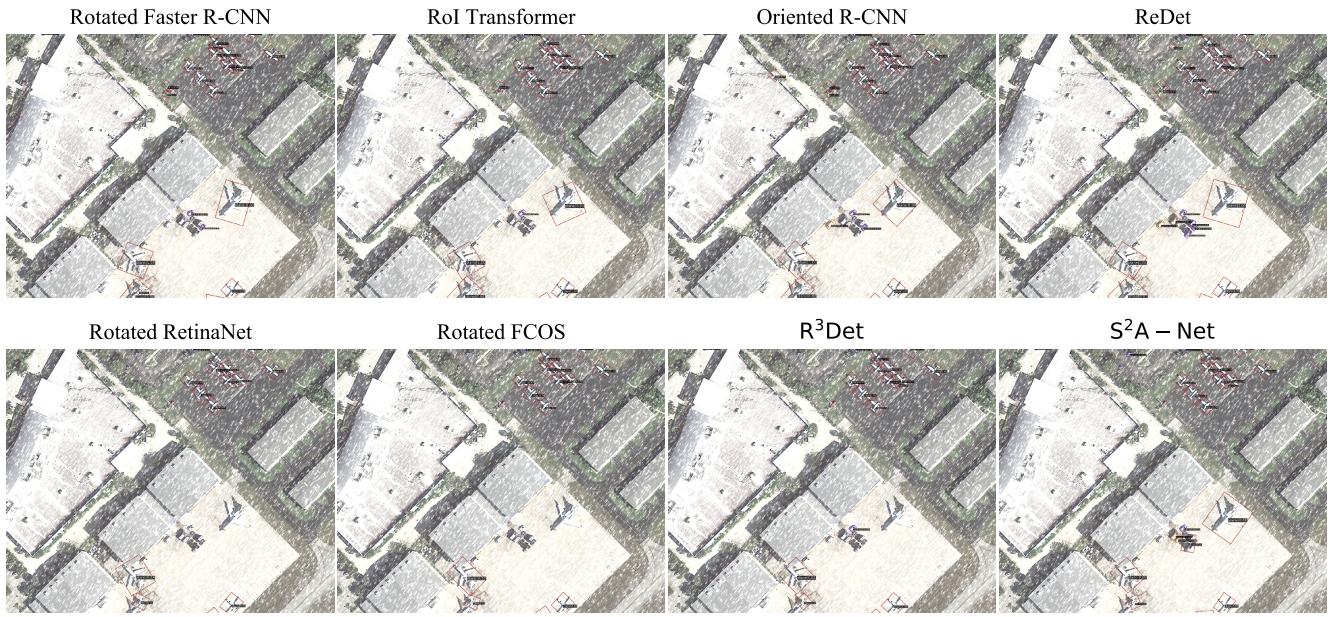


Figure 9. Models' performance on data corrupted with Snow at severity level 3.



Figure 10. Models' performance on data corrupted with Snow at severity level 3. ConvNeXt-Tiny [33], Swin-Transformer-Tiny [32], Swin-Transformer-Small, Swin-Transformer-Base, and Swin-Transformer-Large refer to ROI Transformer [7] with different backbones. RandomRotate and Mosaic [3] refer to two data augmentation strategies used by the model, respectively.

Model	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
Rot. Faster [41]	54.2	42.5	41.6	36.6	48.0	62.7	67.6	56.5	15.8	69.7	40.8	60.9	56.5	72.4	72.4	61.8	22.5	68.3	64.7	67.5
RoI Trans. [7]	55.9	43.8	44.0	39.5	50.1	64.1	70.2	59.5	16.6	71.8	44.0	61.7	56.5	74.5	75.0	63.5	22.0	68.5	66.2	70.7
Ori. R-CNN [52]	56.3	45.5	45.5	38.6	51.7	63.7	69.8	58.9	17.0	72.0	41.8	63.9	58.7	74.6	74.7	63.5	23.5	69.8	66.0	70.2
ReDet [17]	59.9	47.2	48.2	44.1	55.5	69.6	71.8	60.6	19.9	74.7	49.6	68.9	69.4	76.3	75.7	71.0	22.8	73.5	65.9	73.5
Rot. Retina. [30]	50.9	40.0	39.3	32.4	45.2	59.8	64.1	54.2	15.5	66.4	37.8	58.3	54.5	67.5	67.7	58.0	20.9	62.8	60.2	62.8
Rot. FCOS [47]	52.7	40.9	41.0	35.1	46.9	60.4	65.8	55.2	15.8	67.6	40.7	59.2	57.1	69.7	70.6	61.6	19.4	65.1	62.6	67.3
R <sup>3</sup> Det [54]	51.9	41.1	40.6	35.1	46.7	58.8	63.9	53.3	15.8	66.5	39.8	59.4	54.9	68.7	69.1	58.9	21.8	65.6	60.6	64.8
S <sup>2</sup> A-Net [16]	54.5	41.3	41.0	34.3	48.4	62.2	68.0	57.2	16.8	70.9	43.3	62.8	59.6	72.7	72.9	63.7	20.4	66.5	63.9	69.8

Table 11. Two-Stage models: Rotated Faster R-CNN [41], RoI Transformer [7], Oriented R-CNN [52], and ReDet [17]; One-Stage models: Rotated RetinaNet [30], Rotated FCOS [47], R<sup>3</sup>Det [54], and S<sup>2</sup>A-Net [16]. This table shows models' AP<sub>50</sub> on corrupted data of level 1.

Model	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
Rot. Faster [41]	45.4	30.7	28.9	25.1	41.5	55.1	60.7	48.4	12.7	58.3	26.3	49.3	51.4	69.5	52.8	55.0	8.5	66.5	57.7	63.9
RoI Trans. [7]	46.7	31.8	31.2	26.5	44.5	55.7	62.9	50.1	13.3	58.9	28.3	50.1	51.7	71.2	55.1	54.7	9.2	67.5	58.5	65.9
Ori. R-CNN [52]	47.1	33.8	32.7	27.3	44.9	55.3	63.5	50.6	13.6	58.8	28.0	51.9	53.1	71.4	54.2	55.4	8.9	68.7	58.1	65.1
ReDet [17]	52.3	37.3	36.3	32.0	48.2	63.9	66.9	54.0	15.6	66.9	36.6	61.0	65.0	74.3	58.6	66.9	8.9	72.2	59.7	69.0
Rot. Retina. [30]	43.2	30.2	28.7	23.9	39.4	53.2	58.2	46.3	12.3	56.0	25.4	48.2	50.4	64.9	48.2	51.3	9.6	61.2	53.3	59.4
Rot. FCOS [47]	44.8	30.8	29.1	25.6	40.2	53.0	59.5	46.5	12.2	56.7	28.4	49.8	52.7	67.2	52.2	55.6	8.4	64.4	55.5	64.0
R <sup>3</sup> Det [54]	43.7	30.9	29.3	24.8	40.5	51.5	57.1	44.8	12.8	54.8	26.8	49.7	50.7	66.2	49.0	52.8	9.9	63.7	53.6	61.7
S <sup>2</sup> A-Net [16]	45.8	28.5	26.8	22.0	40.5	54.9	61.1	48.5	13.4	58.2	30.2	53.2	55.6	69.8	53.6	57.7	10.2	64.3	56.5	65.7

Table 12. Models' AP<sub>50</sub> on corrupted data of level 2.

Model	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
Rot. Faster [41]	39.2	16.9	16.7	16.8	22.8	37.5	40.5	34.6	15.0	42.3	24.8	42.2	47.1	65.0	42.6	45.4	55.5	54.0	52.7	71.9
RoI Trans. [7]	40.1	15.9	17.1	16.4	24.2	38.8	43.9	36.1	15.1	42.2	26.9	42.7	47.2	66.1	44.4	45.0	57.7	53.4	54.3	74.7
Ori. R-CNN [52]	41.0	19.0	18.8	18.5	26.3	39.5	43.6	35.9	15.7	42.9	25.5	44.0	50.0	68.2	44.3	46.2	57.5	56.7	53.0	74.3
ReDet [17]	47.4	24.1	23.9	23.7	31.4	49.5	50.3	42.8	18.9	54.0	36.2	55.6	61.7	72.1	49.3	58.5	57.3	62.6	54.3	75.0
Rot. Retina. [30]	37.8	17.3	17.5	17.1	22.8	39.5	41.2	34.4	14.9	43.4	23.0	41.0	46.2	61.2	39.6	42.8	53.2	48.4	47.9	67.2
Rot. FCOS [47]	39.5	18.1	18.1	18.7	23.9	40.5	41.9	33.9	15.6	43.5	26.1	42.8	49.3	63.2	41.9	46.6	53.8	52.5	50.7	69.3
R <sup>3</sup> Det [54]	38.0	16.9	16.9	16.9	23.4	36.5	38.5	32.7	15.5	41.2	24.5	42.5	47.1	62.8	39.6	44.1	52.8	53.1	48.7	68.1
S <sup>2</sup> A-Net [16]	39.7	14.1	14.9	13.7	20.4	40.8	43.5	35.1	16.2	43.0	28.6	45.4	51.6	65.7	43.0	48.0	55.5	51.3	51.8	72.1

Table 13. Models' AP<sub>50</sub> on corrupted data of level 3.

Model	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
Rot. Faster [41]	30.9	8.0	7.3	7.3	15.8	27.7	34.4	22.5	13.0	29.7	15.8	41.6	47.1	58.9	38.0	30.5	50.6	41.0	42.5	55.3
RoI Trans. [7]	31.5	6.0	6.0	5.6	16.2	28.0	37.2	24.4	13.3	31.3	17.2	42.9	46.8	58.2	41.0	29.7	53.2	41.9	43.7	56.1
Ori. R-CNN [52]	32.4	7.7	7.1	6.3	17.7	29.4	37.7	24.4	13.0	31.7	17.3	43.5	49.5	61.9	40.9	29.9	52.8	44.1	42.2	59.0
ReDet [17]	38.4	10.6	9.4	9.3	22.7	38.3	43.5	30.4	16.8	43.1	27.8	55.6	61.3	68.7	42.8	43.8	49.9	49.2	45.1	60.9
Rot. Retina. [30]	30.1	8.7	8.0	7.9	15.7	29.0	35.7	22.2	13.2	32.3	15.6	40.9	46.0	55.7	33.3	29.4	47.2	36.6	38.6	55.4
Rot. FCOS [47]	31.4	9.2	9.3	9.6	16.0	30.2	36.9	22.5	13.6	32.6	17.5	42.3	49.2	57.3	36.6	32.1	48.1	40.3	39.7	53.6
R <sup>3</sup> Det [54]	30.4	7.2	7.5	6.6	16.1	26.6	33.5	21.9	13.2	30.3	16.8	42.4	46.8	57.7	34.9	31.4	47.2	41.1	40.2	55.8
S <sup>2</sup> A-Net [16]	32.1	7.2	7.1	6.2	13.7	31.2	38.0	23.0	13.7	33.0	21.7	45.1	51.2	60.1	35.8	33.1	51.0	38.8	42.3	58.1

Table 14. Models' AP<sub>50</sub> on corrupted data of level 4.

Model	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
Rot. Faster [41]	23.7	3.0	3.9	2.4	10.0	19.5	28.8	17.7	13.8	15.2	13.7	37.0	44.4	49.9	27.6	19.5	29.2	35.9	34.6	44.9
RoI Trans. [7]	24.1	1.5	2.9	0.9	10.3	18.7	29.7	18.1	15.3	15.6	16.1	38.1	43.6	47.5	31.4	19.7	32.8	36.5	34.5	44.1
Ori. R-CNN [52]	25.1	2.5	4.4	2.6	11.0	21.5	30.6	18.2	14.9	16.2	15.5	40.2	46.2	51.9	28.4	21.0	31.8	37.9	34.1	48.4
ReDet [17]	30.1	4.2	5.2	2.8	13.5	30.2	35.5	24.7	19.2	27.4	26.1	50.5	58.0	61.2	33.4	31.3	27.2	39.5	36.9	46.0
Rot. Retina. [30]	23.5	4.0	4.8	2.8	10.5	20.8	29.6	17.4	14.3	18.6	14.8	37.4	42.6	47.4	25.7	19.9	26.6	30.9	32.0	45.8
Rot. FCOS [47]	24.6	4.1	4.8	4.6	11.2															

Method	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
ResNet50 [20]	55.9	43.8	44.0	39.5	50.1	64.1	70.2	59.5	16.6	71.8	44.0	61.7	56.5	74.5	75.0	63.5	22.0	68.5	66.2	70.7
ConvNeXt-T [33]	60.3	53.7	54.6	52.6	61.4	66.3	70.5	59.5	19.9	72.7	51.1	67.9	67.7	74.4	74.2	69.5	20.5	71.1	67.8	70.8
Swin-T [32]	59.0	46.6	47.6	48.9	54.1	66.6	71.1	60.3	16.0	74.9	49.5	66.2	64.1	76.7	76.9	68.5	22.1	71.5	67.5	72.5
Swin-S [32]	59.7	50.2	49.4	50.1	57.0	66.0	70.8	60.5	16.7	73.4	51.3	67.1	66.6	75.7	76.3	69.4	23.1	71.5	66.8	73.1
Swin-B [32]	59.8	48.3	48.3	50.1	54.4	67.4	72.4	60.8	16.0	74.9	52.3	67.8	65.9	76.3	76.4	69.9	22.4	71.0	68.9	73.5
Swin-L [32]	62.3	55.1	55.1	55.8	62.4	67.8	71.4	61.9	18.4	76.2	53.1	70.0	69.9	77.1	76.1	73.0	23.0	73.3	69.6	73.8
RandomRotate	57.1	46.8	46.5	40.4	52.2	65.5	71.6	60.4	16.9	73.2	45.1	63.4	59.3	74.8	75.4	64.5	23.0	70.8	65.9	70.1
Mosaic [3]	54.5	44.6	44.1	36.7	49.7	63.2	67.0	57.6	15.3	70.4	40.6	60.0	56.1	73.1	72.9	62.2	21.7	67.0	64.2	69.4

Table 16. ResNet50 [20], ConvNeXt-T (ConvNeXt-Tiny) [33], Swin-T (Swin-Transformer-Tiny) [32], Swin-S (Swin-Transformer-Small), Swin-B (Swin-Transformer-Base), and Swin-L (Swin-Transformer-Large) refer to RoI Transformer with different backbones. Random-Rotate and Mosaic [3] refer to two data augmentation strategies used by the model, respectively. The performance of models' AP<sub>50</sub> on corrupted data of level 1 can be seen in this table.

Method	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
ResNet50 [20]	46.7	31.8	31.2	26.5	44.5	55.7	62.9	50.1	13.3	58.9	28.3	50.1	51.7	71.2	55.1	54.7	9.2	67.5	58.5	65.9
ConvNeXt-T [33]	53.6	46.1	45.5	42.2	55.4	60.0	65.7	53.6	63.9	39.6	59.1	65.1	73.3	60.1	66.8	8.3	67.5	64.3	66.9	
Swin-T [32]	50.4	37.5	36.8	36.8	47.3	59.6	63.8	52.0	13.3	64.1	34.4	55.7	58.8	72.4	61.4	62.4	8.8	61.9	62.2	67.8
Swin-S [32]	51.2	39.1	38.5	37.7	49.8	58.0	64.6	53.3	13.8	61.9	37.2	57.5	61.6	72.4	61.2	62.7	10.3	63.9	61.8	68.4
Swin-B [32]	51.6	39.7	38.9	39.4	49.2	59.9	65.1	53.5	13.1	63.1	37.5	58.6	60.1	73.6	61.1	63.5	10.8	60.2	63.5	69.1
Swin-L [32]	54.0	45.1	43.8	44.3	54.9	59.9	64.1	53.7	14.4	64.2	38.2	61.9	66.2	75.1	61.5	66.3	10.1	66.7	65.3	70.0
RandomRotate	48.2	34.6	34.1	28.6	46.3	57.9	64.4	54.2	13.2	61.5	29.7	50.9	53.9	71.4	54.1	56.7	10.4	69.9	58.1	66.0
Mosaic [3]	46.0	33.4	30.8	26.2	43.5	54.7	60.8	49.1	12.4	59.7	27.5	49.6	51.8	69.5	52.9	55.4	9.6	66.1	56.3	65.3

Table 17. Models' AP<sub>50</sub> on corrupted data of level 2.

Method	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
ResNet50 [20]	40.1	15.9	17.1	16.4	24.2	38.8	43.9	36.1	15.1	42.2	26.9	42.7	47.2	66.1	44.4	45.0	57.7	53.4	54.3	74.7
ConvNeXt-T [33]	49.5	35.9	35.6	36.5	40.9	46.5	46.7	42.9	18.6	49.9	36.1	53.2	63.2	70.4	53.3	60.7	55.2	58.8	61.4	74.0
Swin-T [32]	45.0	24.4	23.5	27.3	32.4	44.7	41.9	37.7	16.5	49.8	34.1	46.9	55.1	69.5	53.0	51.7	58.6	54.8	59.3	74.4
Swin-S [32]	46.0	26.1	24.7	28.0	33.2	45.7	43.1	41.2	16.3	48.1	35.9	49.7	57.3	69.7	52.4	53.0	58.9	57.6	58.9	74.7
Swin-B [32]	46.9	28.2	28.0	31.5	34.5	46.5	45.9	40.3	16.4	49.7	37.1	50.9	55.7	71.0	53.4	52.9	59.8	53.5	60.8	75.8
Swin-L [32]	49.3	33.5	32.8	38.2	39.2	45.7	45.6	40.8	16.5	49.5	35.8	55.4	62.1	72.0	54.6	58.5	60.7	58.2	61.5	76.9
RandomRotate	41.4	17.0	15.8	17.3	25.5	42.0	44.6	39.9	16.2	45.1	28.6	43.1	50.2	67.3	43.5	46.8	57.9	58.0	53.2	74.6
Mosaic [3]	39.3	17.2	16.5	16.6	24.4	39.7	40.4	35.0	14.7	43.6	26.1	42.3	48.5	64.5	41.3	45.4	55.4	51.9	50.9	72.8

Table 18. Models' AP<sub>50</sub> on corrupted data of level 3.

Method	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
ResNet50 [20]	31.5	6.0	6.0	5.6	16.2	28.0	37.2	24.4	13.3	31.3	17.2	42.9	46.8	58.2	41.0	29.7	53.2	41.9	43.7	56.1
ConvNeXt-T [33]	41.2	22.1	19.3	20.6	33.9	37.5	40.8	31.6	16.0	40.2	29.0	52.5	62.3	66.7	48.5	48.6	49.4	46.6	53.7	64.1
Swin-T [32]	35.3	9.2	9.1	8.6	22.3	33.2	33.6	26.0	14.3	37.6	25.0	45.7	54.5	64.4	49.2	34.2	55.3	41.8	50.4	56.3
Swin-S [32]	36.7	11.8	10.3	10.1	23.8	33.5	36.0	28.6	14.6	37.6	27.3	48.8	57.3	63.6	48.9	35.7	55.7	44.8	50.7	57.8
Swin-B [32]	37.4	11.9	10.7	12.0	26.4	36.0	38.3	27.8	13.9	39.1	28.1	49.9	55.7	65.1	49.1	35.2	56.7	42.5	51.4	61.3
Swin-L [32]	40.8	21.3	18.6	23.2	31.7	33.9	38.0	27.8	15.2	37.7	26.3	45.7	62.0	68.4	54.1	42.4	55.0	46.0	53.3	64.5
RandomRotate	32.7	6.3	6.7	5.6	15.9	32.0	38.6	27.4	14.6	33.4	19.6	42.8	50.1	60.8	38.4	29.3	53.9	45.8	42.3	57.9
Mosaic [3]	30.9	8.2	8.1	7.9	15.4	28.6	35.1	23.7	12.6	31.1	18.3	41.8	48.4	57.8	37.3	28.9	49.1	38.5	40.4	56.4

Table 19. Models' AP<sub>50</sub> on corrupted data of level 4.

Method	AP <sub>50</sub>	Noise				Blur				Weather				Digital						
		Ga.	Shot	Im.	Spec.	De.	Glass	Mo.	Zoom	Ga.	Snow	Frost	Fog	Br.	Spat.	Co.	El.	Pixel	JPEG	Sa.
ResNet50 [20]	24.1	1.5	2.9	0.9	10.3	18.7	29.7	18.1	15.3	15.6	16.1	38.1	43.6	47.5	31.4	19.7	32.8	36.5	34.5	44.1
ConvNeXt-T [33]	32.8	8.7	10.7	7.7	24.2	28.9	36.5	25.7	18.6	27.0	29.4	48.7	58.9	59.8	38.0	34.5	26.6	39.7	45.2	53.4
Swin-T [32]	25.9	3.8	5.6	3.7	14.9	23.2	29.9	19.9	16.1	19.9	19.6	40.6	51.1	55.9	36.3	18.4	34.7	23.4	39.0	36.8
Swin-S [32]	27.7	4.1	6.1	3.9	15.8	24.9	29.2	22.8	16.4	20.8	22.8	44.0	53.9	55.6	36.5	20.6	39.1	31.7	40.4	37.9
Swin-B [32]	28.2	4.7	6.9	4.0	18.1	25.6	32.1	22.8	15.5	21.9	22.7	45.4	52.4	57.3	36.1	20.9	38.2	25.6	42.3	42.7
Swin-L [32]	31.1	7.4	10.4	8.1	22.1	26.5	31.2	23.4	17.6	21.9	25.9	49.8	5							

Method	rPC (%)	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC
ResNet50 [20]	52.12	66.61	42.96	35.41	47.21	47.69	50.91	63.84	75.64	50.11	51.65	43.83	46.04	53.43	50.59	37.72
ConvNeXt-T [33]	63.31	77.90	54.86	47.38	60.39	61.31	66.08	71.49	83.08	56.62	59.74	50.64	56.63	70.36	64.35	52.41
Swin-T [32]	55.64	69.78	46.36	41.93	53.94	54.90	57.26	66.35	76.81	50.47	50.60	44.27	49.17	60.00	57.65	41.01
Swin-S [32]	57.42	71.28	47.31	41.03	55.17	56.33	60.46	68.27	78.22	51.44	53.43	45.46	53.53	62.50	56.89	45.06
Swin-B [32]	57.64	70.09	53.00	40.41	53.11	54.41	59.79	69.09	79.31	53.01	52.22	46.00	48.57	63.59	59.62	47.07
Swin-L [32]	61.15	73.43	50.30	44.05	57.49	62.03	71.03	71.84	83.01	55.60	55.07	49.22	55.17	67.62	56.13	52.02
ReDet [17]	59.51	71.54	54.42	42.49	56.19	56.97	62.42	70.92	79.78	57.16	59.02	50.69	51.98	63.36	60.41	37.46
RandomRotate	53.59	67.88	44.58	39.98	49.60	50.61	53.76	65.67	74.56	49.93	51.14	43.54	47.71	55.86	53.42	39.34
Mosaic [3]	52.23	65.26	40.75	41.14	48.37	48.46	53.36	66.18	71.71	46.71	51.91	39.87	43.64	56.03	51.80	41.91

Table 21. Models' rPC for each object category. From left to right, the types of objects are plane, baseball diamond, bridge, ground track field, small vehicle, large vehicle, ship, tennis court, basketball court, storage tank, soccer ball field, roundabout, harbor, swimming pool, and helicopter.

Method	rPC <sub>clouds</sub> (%)	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC
ResNet50 [20]	78.90	98.71	61.90	58.00	70.95	80.53	83.25	90.34	99.85	77.18	83.62	64.57	73.54	55.02	85.81	83.04
ConvNeXt-T [33]	86.04	98.66	74.12	64.98	80.87	88.01	89.47	90.79	99.89	86.33	87.72	75.02	79.84	81.02	88.82	89.15
Swin-T [32]	81.06	98.19	62.03	60.35	76.04	81.59	87.27	90.61	99.87	79.06	86.96	65.92	74.02	74.69	87.69	74.52
Swin-S [32]	82.05	90.92	69.62	60.26	77.68	82.28	89.08	90.74	99.85	80.02	80.39	67.17	81.78	76.23	88.22	83.86
Swin-B [32]	83.45	98.15	74.71	59.62	73.68	81.66	88.35	90.91	99.88	86.41	86.34	66.77	72.91	76.11	88.62	91.74
Swin-L [32]	85.94	99.15	70.33	63.71	77.91	87.75	97.20	90.76	99.89	87.05	88.20	69.51	75.07	86.32	87.31	95.89
ReDet [17]	86.33	99.29	72.78	65.68	78.51	83.12	88.36	98.19	99.94	88.17	93.36	73.22	77.62	84.41	96.17	78.94
RandomRotate	80.21	98.58	65.95	60.63	71.80	80.49	81.32	90.56	99.86	81.99	84.61	56.92	71.78	64.58	86.65	89.12
Mosaic [3]	80.10	98.26	60.02	61.96	74.25	83.79	83.94	90.88	99.68	77.91	85.48	50.44	76.64	66.77	84.92	85.27

Table 22. Models' rPC<sub>clouds</sub> for each object category.