2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Tiny Object Detection in Aerial Images

Jinwang Wang[1], Wen Yang[1]*, Haowen Guo[1], Ruixiang Zhang[1], Gui-Song Xia[2]
[1]School of Electronic Information, [2]School of Computer Science
Wuhan University, Wuhan, Hubei, China
{jwwangchn, yangwen, ghw, zhangruixiang, guisong.xia}@whu.edu.cn

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9413340

***Abstract*—Object detection in Earth Vision has achieved great progress in recent years. However, tiny object detection in aerial images remains a very challenging problem since the tiny objects contain a small number of pixels and are easily confused with the background. To advance tiny object detection research in aerial images, we present a new dataset for Tiny Object Detection in Aerial Images (AI-TOD). Specifically, AI-TOD comes with 700,621 object instances for eight categories across 28,036 aerial images. Compared to existing object detection datasets in aerial images, the mean size of objects in AI-TOD is about 12.8 pixels, which is much smaller than others. To build a benchmark for tiny object detection in aerial images, we evaluate the state-of-the-art object detectors on our AI-TOD dataset. Experimental results show that direct application of these approaches on AI-TOD produces suboptimal object detection results, thus new specialized detectors for tiny object detection need to be designed. Therefore, we propose a multiple center points based learning network (M-CenterNet) to improve the localization performance of tiny object detection, and experimental results show the significant performance gain over the competitors.**

***Index Terms*—tiny object detection, aerial image, benchmark, convolutional neural network**

I. Introduction

Object detection in aerial image is an open issue, with wide applications including large-scale surveillance, intelligent transportation, and location-based service [1]–[3]. Although recent investigation for object detection problem has already achieved great progress, it is still a very challenging problem when objects in aerial images are microscopic (*e.g.*, tiny vehicles with less than 8 pixels) [4], [5].

Unlike objects in proper scales, detecting objects of tiny scale is much more challenging due to the extremely small size and low signal-to-noise ratio in aerial image [6]. For the convolutional neural network (CNN) based object detection methods such as Faster R-CNN [7] with ResNet-50 [8], input images will be downsampled 16 times by pooling layers. Therefore, a number of tiny objects will be filtered out in the final feature map. Despite the fact that lots of methods tackled this problem [9]–[13], as shown in recent work about the upper bound of object detection performance [14], there is still a large gap between current and the upper bound performances on tiny object detection.

To obtain better performance on tiny object detection, we not only need to design special detectors, but also need to establish a specialized benchmark. However, existing large-scale object detection benchmarks like DOTA [1] and DIOR [3] contain objects of various scales. The largest object in DOTA and DIOR are $1,698$ pixels and 764 pixels, respectively. In addition, as shown in Fig. 1, there are 67.8% and 79.0% of objects larger than 16 pixels in DOTA and DIOR datasets, respectively. Therefore, these datasets are not suitable for applications like tiny object detection and recognition. In a typical aerial image with two meters' space resolution, ordinary objects like vehicles are usually smaller than 8 pixels, which further increases the difficulty of tiny object detection.

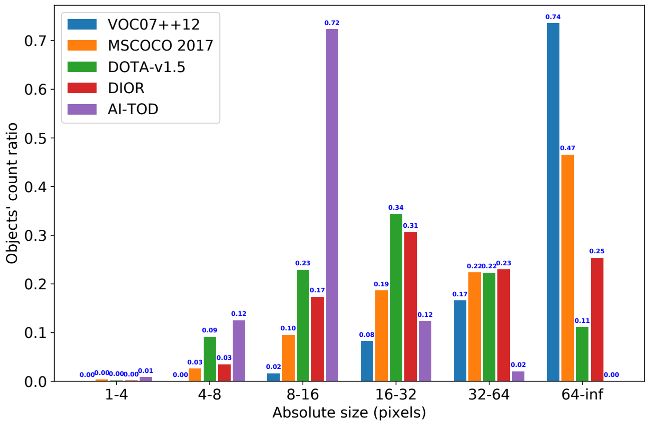


Fig. 1. Comparison of AI-TOD with other benchmark datasets. The largest object in AI-TOD is smaller than 64 pixels, and 86% of objects in AI-TOD are smaller than 16 pixels.

In addition, due to these datasets containing a number of large objects, they are not suitable for evaluating the performance of detectors on tiny object detection. To benchmark typical detectors on the tiny object detection task fairly, a tiny object detection dataset in aerial images is established, which is called as AI-TOD. The AI-TOD contains 700,621 object instances for eight categories across 28,036 aerial images. Unlike the aforementioned datasets, the largest object in AI-TOD is smaller than 64 pixels, and 86% of objects in AI-TOD are smaller than 16 pixels as illustrated in Fig. 1. Notably, as shown in Tab. II, the mean size of objects in AI-TOD is 12.8 pixels, which is much smaller than that in both aerial images and natural image detection datasets.

Furthermore, we propose a simple yet effective approach, named multiple center points based learning network (M-CenterNet), to detect tiny objects in aerial images. The intuition of our approach is first to locate the multiple center points, then estimate the multiple offsets and scale of the corresponding object. In our experiments, the proposed M-CenterNet can improve the localization performance and gain significant performance in AI-TOD on both AP and oLRP [15] metrics.

The main contributions of this paper are:

- We introduce AI-TOD, a dataset for tiny object detection in aerial images. Besides, we establish a corresponding benchmark by several CNN-based object detectors and provide an overview of the state-of-the-art detectors on the AI-TOD dataset. The training/validation images and annotations will be made public, and an online benchmark will be set up for algorithm evaluation.
- We propose a multiple center points based learning network (M-CenterNet) for tiny object detection, it obtains state-of-the-art performance in the AI-TOD dataset on both AP and oLRP metrics.

II. Related Work

A. Dataset for Aerial Object Detection

In the past years, numerous object detection datasets in aerial images, such as NWPU VHR-10 [16], HRSC2016 [17], VEDAI [18], xView [19], DOTA [1], VisDrone [20], UAVDT [21], and DIOR [3] have been proposed to make advancement in object detection research for Earth Vision. However, the objects in these datasets have a variety of scales, which leads to the result that these datasets are more suitable for evaluating the detectors designed for multi-scale object detection rather than tiny object detection. Although some works on tiny object detection in aerial images use aerial datasets (*e.g.*, VEDAI [18] and $\mathcal{R}^2$-CNN [4]), VEDAI only focuses on vehicle detection, and dataset in $\mathcal{R}^2$-CNN is not publicly available.

Our proposed AI-TOD contains objects with eight categories, and 86% objects are less than 16 pixels. Compared with the aforementioned datasets, AI-TOD is more suitable for evaluating the performance of multi-categories tiny object detection. Furthermore, AI-TOD is publicly available to compare the performance of object detectors.

B. Object Detection in Aerial Images

Compared with handcrafted feature-based object detection methods, CNN-based object detection methods have shown remarkable improvements in both accuracy and speed recently. CNN-based detectors can be categorized into two main categories: anchor-based and anchor-free detectors [22]. The former can be divided into one-stage and two-stage detectors, while the latter falls into keypoint-based and center-based detectors [22].

Anchor-based Detector: For two-stage detectors, the most representative work is Faster R-CNN [7], which consists of a region proposal network (RPN) and a region-wise prediction network (R-CNN) [23] to detect objects. After that, lots of detectors are proposed to improve its performance, including FPN [9], Cascade R-CNN [24], and Trident-Net [10]. For one-stage detectors, they directly predict class probabilities and bounding box offsets like SSD [25], RetinaNet [26] and YOLOv3 [27]. Therefore, one-stage detectors are simpler and more efficient than two-stage detectors.

Anchor-free Detector: For keypoint-based detectors, they first locate several pre-defined or self-learned keypoints, and then generate bounding boxes to detect objects [22], such as CornerNet [28], Grid R-CNN [29], CenterNet [30], and RepPoints [31]. For center-based detectors, they regard the center of object as foreground to define positives, and then predict the distances from the positives to the four sides of the object bounding box for detection [22], like DenseBox [32], FCOS [33], and FoveaBox [34].

Inspired by the great success of CNN-based object detectors in natural scenes, extensive studies recently focus on object detection in aerial images. Unlike object detectors in natural scenes, most of the studies use anchor-based detectors to detect objects in aerial images, such as R-P-Faster R-CNN [35], YOLT [36], RoI Transformer [37], SCRDet [38], $\mathcal{R}^2$-CNN [4], and Mask OBB [2].

C. Tiny Object Detection in Aerial Images

Small or tiny object detection is a very challenging topic, and researchers have proposed some methods for tiny object detection. SSD [25] tackles tiny object detection problem by increasing the input image resolution. FPN [9] fuses features of different levels via the top-down pathway and the lateral connection to detect small objects. PSPNet [39] proposes a pyramid scene parsing network which employs the context information to solve tiny object detection problem. Sig-NMS [5] proposes a new NMS method for improving the detection accuracy of tiny objects in aerial images. $\mathcal{R}^2$-CNN [4] proposes a specially designed backbone named Tiny-Net to detect tiny objects in large-scale aerial images. Yu *et al.* [6] propose a Scale Match method for tiny object detection by aligning object scales between two datasets.

Our proposed M-CenterNet is an anchor-free keypoint-based detector, and it uses multiple center points to locate accurate object center for improving the location performance of tiny object detection.

III. Dataset Details

A. Dataset Construction Process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5 `trainval set` [1], xView `training set` [19], VisDrone2018-Det `trainval set` [20], Airbus Ship `trainval set`[1] and DIOR `trainval+test sets` [3]. The details of these datasets are as follows:

DOTA-v1.5 `trainval set`: It is an upgraded version of original DOTA-v1.0 dataset [1], and has been employed for performance evaluation in *Detecting Objects in Aerial Images (DOAI2019)*[2]. DOTA-v1.5 `trainval set` contains $1,869$ images ranging in size from 800×800 to 4000×4000 pixels and $280,196$ object instances which are labeled into 16 classes (*e.g.*, *ship, small-vehicle, storage-tank*).

xView `training set`: It is a large-scale object detection dataset which consists of $1,415\text{km}^2$ of WorldView-3 at 30cm

[1] https://www.kaggle.com/c/airbus-ship-detection

[2] https://captain-whu.github.io/DOAI2019/

resolution. The annotated dataset for object detection contains over 1 million object instances across 60 classes, including various types of vehicles, planes, and ships [19].

VisDrone2018-Det `trainval set`. It consists of 7,019 images captured by drone platforms in different places at different heights [20]. Images are manually annotated with bounding boxes and 10 predefined classes (*e.g.*, *pedestrian, person, car*).

Airbus-Ship `trainval set`. It is a dataset for ship detection in Kaggle challenge. Airbus-ship `trainval set` consists of $42,559$ images and $81,724$ ships, all objects are annotated by polygon.

DIOR `trainval+test set`. It includes $23,463$ images and $192,472$ object instances covered by 20 classes (*e.g.*, *airplane, ship, windmill*).

To establish the AI-TOD, we extract images and object instances from the above datasets as follows:

1) **Image size.** Original images are divided into 800×800 patches with an overlap of 200 pixels. If the original images are smaller than 800×800 pixels, it will be padded to 800×800 pixels by zero pixels.
2) **Object type.** Eight categories are chosen in our AI-TOD dataset, including *airplane (AI), bridge (BR), storage-tank (ST), ship (SH), swimming-pool (SP), vehicle (VE), person (PE), wind-mill (WM)*. The categories are selected according to whether a kind of object is common and its size in low-resolution aerial images.
3) **Category conversion.** After the selection of categories, we convert the old categories in the corresponding dataset to new categories. In this process, several objects whose categories are not contained within AI-TOD will be dropped.
4) **Image selection.** The images are chosen by the proportion of tiny objects and the number of large objects in the images. Note that the size of the object in this work is defined as the square root of the objects bounding box area. Specifically, absolute size $S_a(\cdot)$ and relative size $S_r(\cdot)$ of an object can be calculated by [6]:

$$S_a(b_i) = \sqrt{w_i \times h_i}, \tag{1}$$

$$S_r(b_i) = \sqrt{\frac{w_i \times h_i}{W \times H}}, \tag{2}$$

where $b_i = (cx_i, cy_i, w_i, h_i)$ denotes the bounding box of the i-th object in image I, (cx_i, cy_i), w_i and h_i are the center coordinates, width and height of b_i, W and H are the width and height of the image. Note that W and H are both 800 in AI-TOD. Therefore, the size set of bounding boxes in image I can be represented as $S_a(I) = \{S_a(b_1), S_a(b_2), \dots S_a(b_N)\}$, where N is the number of bounding boxes in image I. Then, the number of tiny object N_t and large object N_l can be defined as:

$$N_t = \sum_{i=1}^{N} \mathbb{1}_A(S_a(b_i)), N_l = \sum_{i=1}^{N} \mathbb{1}_B(S_a(b_i)), \tag{3}$$

where $\mathbb{1}_A(\cdot)$ and $\mathbb{1}_B(\cdot)$ are the indicator functions. In this work, they are defined as:

$$\mathbb{1}_A(x) = \begin{cases} 1 & \text{if } x \leq 16, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

$$\mathbb{1}_B(x) = \begin{cases} 1 & \text{if } x \geq 64, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Besides, for keeping more tiny objects and dropping large objects, we keep the image when $N_t/N > 50\%$ and $N_l = 0$.

After the processing mentioned above, we obtain the final tiny object detection dataset AI-TOD, which comes with $700,621$ object instances for eight categories across $28,036$ aerial images[3] with sizes of 800×800 pixels.

For dataset splits, $2/5$, $1/10$ and $1/2$ of the images are used to form `training set`, `validation set` and `test set`. For each object category and image set, the number of object instances is reported in Tab. I. Both images and annotations for the `training` and `validation sets` will be publicly available. In the case of `test set`, we will publicly only provide images without annotations. The `test set` annotations will be used to set up an evaluation server for a fair comparison between the detectors.

TABLE I
NUMBER OF OBJECTS PER IMAGE SET AND PER CLASS.

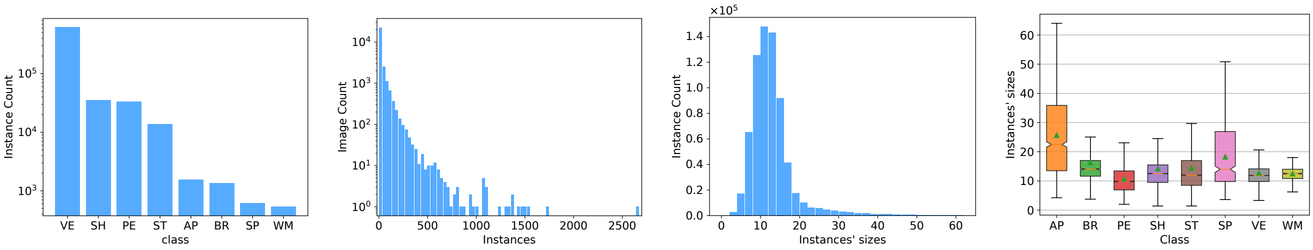
AI-TOD	Train	Validation	Trainval	Test
airplane (AI)	623	170	793	745
bridge (BR)	512	140	652	689
storage-tank (ST)	5,269	2,477	7,746	5,860
ship (SH)	13,539	3,791	17,330	17,633
swimming-pool (SP)	293	34	327	292
vehicle (VE)	248,042	59,904	307,946	306,665
person (PE)	14,126	3,841	17,967	15,443
wind-mill (WM)	176	67	243	290
Total	282,580	70,424	353,004	347,617

B. AI-TOD Statistics

In this section, the properties of AI-TOD are analyzed and compared with other relevant datasets.

Object count per class. AI-TOD contains $700,621$ annotated object instances of eight categories. Fig. 2a shows that there are some infrequent classes (*e.g.*, *swimming-pool (SP), wind-mill (WM)*) with significantly less number of objects than other more frequent classes (*e.g.*, *vehicle (VE), ship (SH)*). Such a class imbalance usually exists in aerial image datasets (*e.g.* DOTA [1], DIOR [3]) and it is important for real-world applications.

[3] The copyrights of the images in AI-TOD belongs to the authors of the corresponding original dataset, while AI-TOD is designed only for academic research.



(a) (b) (c) (d)

Fig. 2. Statistics of classes and instances in AI-TOD. (a) Histogram of the number of instances per class. (b) Histogram of number of instances per image. (c) Histogram of number of instances' sizes. (d) Boxplot depicting the range of sizes for each object category. Note that short name are used to define categories.

TABLE II
MEAN AND STANDARD DEVIATION OF OBJECT SCALE ON DIFFERENT DATASETS.

Dataset	Absolute size (pixels)	Relative size (pixels)
PASCAL VOC 07++12	156.6±111.2	0.372±0.265
MS COCO `trainval`	99.5±107.5	0.190±0.203
xView	34.9±39.9	0.011±0.013
DOTA-v1.0 `trainval`	55.3±63.1	0.028±0.034
DOTA-v1.5 `trainval`	34.0±47.8	0.016±0.026
VisDrone	35.8±32.8	0.030±0.026
Airbus-Ship	44.9±44.1	0.058±0.057
DIOR	65.7±91.8	0.082±0.115
AI-TOD	**12.8±5.9**	**0.016±0.007**

Object count per image. Due to a large field of view, there are many interesting objects in one aerial image. As shown in Fig. 2b, the object count per image in AI-TOD can reach up to $2,667$, which is much more than normal object detection datasets whether in natural images or aerial images.

Object size distribution. Fig. 2c shows the size distribution of AI-TOD. The object size is mainly located around 12 pixels. As shown in Tab. II, the mean and standard deviation of the absolute size of AI-TOD is 12.8 pixels and 5.9 pixels, respectively, which are much smaller than other natural image and aerial image datasets.

Object size of categories. In AI-TOD, objects appear in various sizes, and we consider objects in the range 2 to 8 pixels as *very tiny*, 8 to 16 pixels as *tiny*, 16 to 32 as *small*, 32 to 64 as *medium*, and no large objects. The percentages of *very tiny, tiny, small* and *medium* objects in AI-TOD is 13.3%, 72.3%, 12.3% and 2.1%, respectively. The box plot in Fig. 2d presents the statistics of area for each category of AI-TOD.

IV. M-CenterNet for Tiny Object Detection

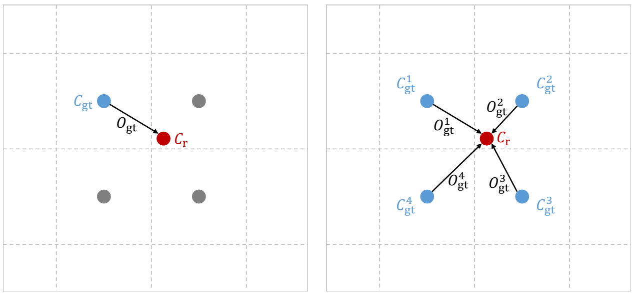
To distinguish tiny objects from the input image, high-resolution feature maps are required. Thus, the keypoint prediction network, which can output high quality and high-resolution feature maps, would be a good choice for tiny object detection. Our proposed M-CenterNet is inspired by the anchor-free keypoint-based detector CenterNet [30], which uses Deep Layer Aggregation (DLA) [40] as the high-resolution feature extraction network. Different from CenterNet, we redesign the center point and offset targets for better tiny object detection.

To ensure satisfactory performance, detectors need to predict high-quality bounding boxes which have high Intersection over Unions (IoUs) with ground truths. However, IoU is very sensitive to tiny objects. For instance, the deviation of one pixel may cause the predicted bounding box to change from the positive to the negative. Therefore, accurate localization ability is necessary to obtain high tiny object detection performance. However, the original CenterNet uses a single center point as the ground truth. Fig. 3a shows the design of the original CenterNet, in which, red circle point is the real center point $C_r = (cx/s, cy/s)$ on feature map, blue point $C_{\text{gt}} = (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor)$ and grey circle points are treated as positive and negative samples in training stage, O_{gt} is the offset of corresponding positive sample, where (cx, cy) is the center point of the object on image, s is the output stride of feature map. In this design, when $(\lfloor cx/s \rfloor, \lfloor cy/s \rfloor) \rightarrow (\lceil cx/s \rceil, \lceil cy/s \rceil)$, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are defined by the set equations in (6), predicted bounding box may have up to four pixels deviation. In this situation, IoU of the predicted bounding box and ground truth might be less than 0.5, and this object will be missed.

$$\begin{aligned} \lfloor x \rfloor &= \max\{m \in \mathbb{Z} | m \leq x\}, \\ \lceil x \rceil &= \min\{n \in \mathbb{Z} | n \geq x\} \end{aligned} \tag{6}$$

To tackle this problem, instead of using a single center point design, we use multiple center points' design. As shown in Fig. 3b, we treat four points around the real center point as positive samples, the points except for the positive samples are the negative samples. Red circle point is the real center point $C_r = (cx/s, cy/s)$ on feature map, and blue points

$$\begin{aligned} C_{\text{gt}}^1 &= (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor), \\ C_{\text{gt}}^2 &= (\lceil cx/s \rceil, \lfloor cy/s \rfloor), \\ C_{\text{gt}}^3 &= (\lfloor cx/s \rfloor, \lceil cy/s \rceil), \\ C_{\text{gt}}^4 &= (\lceil cx/s \rceil, \lceil cy/s \rceil) \end{aligned}$$



(a) Single center point (b) Multiple center points

Fig. 3. Illustration for the center and offset ground truth for training detector. (a) Single center point design. Red circle point is the real center point on the feature map, blue point and grey circle points are treated as positive and negative samples in training period, O_{gt} is the offset of corresponding positive. (b) Multi center points design. Red circle point is the real center point on feature map, blue points are treated as positive samples in training period, $\{O^i_{\mathrm{gt}}, i = 1, 2, 3, 4\}$ are the offsets of corresponding positive samples.

are treated as positive samples in training period, and

$$\begin{aligned} O^1_{\mathrm{gt}} &= (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor), \\ O^2_{\mathrm{gt}} &= (\lceil cx/s \rceil, \lfloor cy/s \rfloor), \\ O^3_{\mathrm{gt}} &= (\lfloor cx/s \rfloor, \lceil cy/s \rceil), \\ O^4_{\mathrm{gt}} &= (\lceil cx/s \rceil, \lceil cy/s \rceil) \end{aligned}$$

are the offsets of corresponding positive samples. We call this new detector as Multiple Center Points based Learning Network (M-CenterNet). Besides the ground truth for training detector, the losses of center points, offsets, and sizes need to be calculated four times by the loss functions of original CenterNet and taken the average values as final losses. In the inference stage, different from CenterNet, we use 2×2 average pooling instead of 3×3 max pooling to find the center points from the feature map, and we use non maximum suppression (NMS) to filter redundant bounding boxes.

V. Experiments

A. Experimental Setup

Implementation Details: We conducted all experiments on a computer with 4 NVIDIA Titan X GPUs. Deep learning-based object detectors, which are widely used for object detection in natural images, are selected as our benchmark testing algorithms. Specifically, our selections include

- *anchor-based two-stage detectors:* Faster R-CNN [7], Cascade R-CNN [24] and TridentNet [10];
- *anchor-based one-stage detectors:* YOLOv3 [27], RetinaNet [26], SSD [25];
- *anchor-free center-based detectors*: FoveaBox [34] and FCOS [33];
- *anchor-free keypoint-based detectors:* RepPoints [31], CenterNet [30] and M-CenterNet.

For Faster R-CNN, Cascade R-CNN, RetinaNet, SSD, FoveaBox, FCOS, and RepPoints, the codes are based on MMDetection [41] library[4]. For TridentNet[5], YOLOv3[6] and CenterNet[7], we use the official code. Note that we keep all the experimental settings the same as that depicted in code libraries.

Besides, the backbone networks are ResNet-50 [8] for Faster R-CNN, Cascade R-CNN, TridentNet, RetinaNet, FoveaBox, FCOS and RepPoints, VGG-16 [42] for SSD, DarkNet-53 [27] for YOLOv3, and DLA-34 for CenterNet and M-CenterNet.

Evaluation Metrics: We employ two metrics to evaluate the detection performance on AI-TOD quantitatively. One is the Average Precision (AP) metric, which has been widely used to assess various detection algorithms. However, the localization performance is critical since the IoU between objects is very sensitive when objects are tiny. Besides, the common AP computation considers localization accuracy by an indirect way (by the IoU computation for the True Positive (TP), False Positive (FP), and False Negative (FN)). Therefore, we employ the other metric called the Optimal Localization Recall Precision (oLRP) [15] to obtain a more reliable evaluation on localization performance. The oLRP of ground truth boxes X and detection boxes Y_s with confidence score larger than a score threshold $s \in [0, 1]$ can be defined as follows:

$$\begin{aligned} \mathrm{oLRP}(X, Y_s) = \min_s \frac{1}{Z} \Big(& \frac{N_{\mathrm{TP}}}{1-\tau} \mathrm{LRP}_{\mathrm{IoU}}(X, Y_s) \\ & + |Y_s| \mathrm{LRP}_{\mathrm{FP}}(X, Y_s) \\ & + |X| \mathrm{LRP}_{\mathrm{FN}}(X, Y_s) \Big), \end{aligned} \tag{7}$$

where

$$\mathrm{LRP}_{\mathrm{IoU}}(X, Y_s) = \frac{1}{N_{\mathrm{TP}}} \sum_{i=1}^{N_{\mathrm{TP}}} (1 - \mathrm{IoU}(x_i, y_{x_i})), \tag{8}$$

$$\mathrm{LRP}_{\mathrm{FP}}(X, Y_s) = 1 - \mathrm{Precision} = \frac{N_{\mathrm{FP}}}{|Y_s|}, \tag{9}$$

$$\mathrm{LRP}_{\mathrm{FN}}(X, Y_s) = 1 - \mathrm{Recall} = \frac{N_{\mathrm{FN}}}{|X|}, \tag{10}$$

$\tau \in [0, 1)$ is the IoU threshold. $\mathrm{IoU}(x_i, y_{x_i})$ denotes the IoU between $x_i \in X$ and its assigned detection $y_{x_i} \in Y_s$. $N_{\mathrm{TP}}, N_{\mathrm{FP}}, N_{\mathrm{FN}}$ are the number of TP, FP and FN, respectively. $Z = N_{\mathrm{TP}} + N_{\mathrm{FP}} + N_{\mathrm{FN}}$.

Note that AP is a higher-is-better measure, while oLRP is an error metric, and thus, it is a lower-is-better measure. In addition, AP_{vt}, AP_t, AP_s, AP_m are APs for *very tiny, tiny, small, medium* scales, respectively.

B. Experimental Results

In Tab. III and IV, we report the results achieved by baseline methods. It can be seen that the whole performance is much lower than the performance of these methods on the MS COCO [43] and PASCAL VOC [44] datasets, which implies these methods can not be well used on the real-world's

[4] https://github.com/open-mmlab/mmdetection
[5] https://github.com/TuSimple/simpledet
[6] https://github.com/pjreddie/darknet
[7] https://github.com/xingyizhou/CenterNet

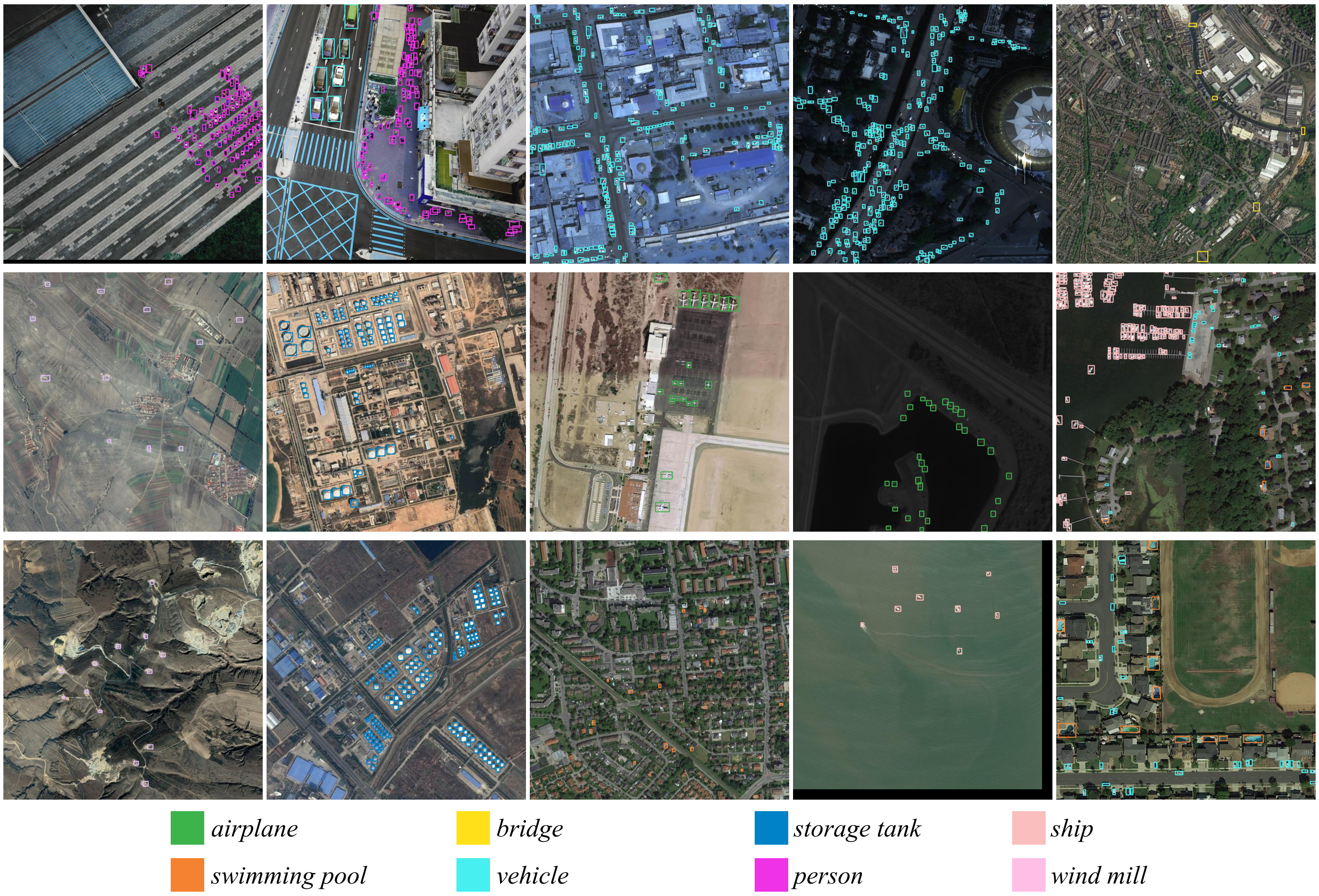


Fig. 4. Samples of annotated images in AI-TOD. Best viewed in color and zoomed in.

tiny object detection in aerial images. Therefore, a specially designed benchmark (AI-TOD) for tiny object detection in aerial images is essential.

Generally, *anchor-free keypoint-based* detectors achieve the better performance since these detectors do not use IoU to assign positive and negative samples, while tiny objects are sensitive to IoU. Some detectors like TridentNet, YOLOv3, and RetinaNet can not be easily adapted to AI-TOD well, which can obtain excellent performance in the MS COCO dataset. For $\mathrm{AP}_{0.75}$, which is a metric that requires high localization performance, some detectors such as YOLOv3, RetinaNet, SSD-512 and TridentNet are less than 5.0%, which implies poor localization performance of these detectors. For $\mathrm{AP}_{\mathrm{vt}}$, most of the detectors in the benchmark can just obtain performance less than 3%, which actually can not be used in the real-word's applications.

In addition, thanks to the high-resolution feature map and great localization performance, M-CenterNet obtains the best performance in seven metrics as shown in Tab. III. Especially, in the $\mathrm{AP}_{\mathrm{vt}}$ and AP_{t} metrics, M-CenterNet far surpasses other detectors. Besides, as shown in Tab. IV, M-CenterNet obtains the best performances on five categories (*bridge (BR), storage-tank (ST), vehicle (VE), person (PE), and wind-mill (WM)*).

VI. Conclusion

We have built a dataset for tiny object detection in aerial images in which the mean object size is much smaller than existing object detection datasets. We benchmark AI-TOD using several popular object detection approaches, including four categories (anchor-based two-stage, anchor-based one-stage, anchor-free center-based, anchor-free keypoint-based), twelve detectors. Experimental results show that direct application of these approaches on AI-TOD provides suboptimal object detection results, thus requiring specialized solutions. In addition, we propose a multiple center points based learning network (M-CenterNet) to improve the localization performance of tiny object detection, and experimental results show the significant performance gain over the state-of-the-art detectors. We believe AI-TOD will not only promote the development of tiny object detection algorithms in Earth Vision, but also provide another evaluation perspective for general multi-scale object detection algorithms. Our codes and dataset are available at: https://github.com/jwwangchn/AI-TOD.

Acknowledgment

The work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant 61771351

TABLE III
PERFORMANCE OF TWELVE DETECTORS ON THE PROPOSED AI-TOD `test set`. BOLD AND UNDERLINE FONTS INDICATE THE BEST AND THE SECOND BEST PERFORMANCES FOR EACH METRIC, RESPECTIVELY.

Method	Backbone	AP	$AP_{0.5}$	$AP_{0.75}$	AP_{vt}	AP_{t}	AP_{s}	AP_{m}	oLRP	$oLRP_{IoU}$	$oLRP_{FP}$	$oLRP_{FN}$
anchor-based two-stage:												
TridentNet [10]	ResNet-50	7.5	20.9	3.6	1.0	5.8	12.6	14.0	92.7	33.3	60.0	72.6
Faster R-CNN [7]	ResNet-50-FPN	11.4	27.0	8.0	0.0	8.3	23.1	24.5	89.5	29.9	49.2	71.1
Cascade R-CNN [24]	ResNet-50-FPN	13.8	30.8	**10.5**	0.0	10.6	**25.5**	**26.6**	87.6	**27.2**	45.1	68.6
anchor-based one-stage:												
YOLOv3 [27]	DarkNet-53	4.5	14.2	1.7	2.1	4.6	5.9	6.2	94.3	33.7	44.8	80.4
RetinaNet [26]	ResNet-50-FPN	4.7	13.6	2.1	2.0	5.4	6.3	7.6	94.7	33.0	74.4	78.2
SSD-512 [25]	VGG-16	7.0	21.7	2.8	1.0	4.7	11.5	13.5	92.8	33.5	60.4	71.1
anchor-free center-based:												
FoveaBox [34]	ResNet-50-FPN	8.1	19.8	5.1	0.9	5.8	13.4	15.9	92.6	27.2	57.9	79.4
FCOS [33]	ResNet-50-FPN	9.8	24.1	5.9	1.4	8.0	15.1	17.4	90.8	29.6	56.4	73.4
anchor-free keypoint-based:												
RepPoints [31]	ResNet-50-FPN	9.2	23.6	5.3	2.5	9.2	12.9	14.4	91.5	29.5	58.2	75.0
Grid R-CNN [29]	ResNet-50-FPN	12.2	27.7	9.0	0.2	10.3	22.6	23.3	88.6	28.3	48.8	70.6
CenterNet [30]	DLA-34	13.4	39.2	5.0	3.8	12.1	17.7	18.9	87.1	32.7	41.8	56.9
M-CenterNet	DLA-34	**14.5**	**40.7**	6.4	**6.1**	**15.0**	19.4	20.4	**85.8**	31.5	**39.3**	**54.8**

TABLE IV
CLASS-WISE OBJECT DETECTION RESULTS ON AI-TOD `test set`. BOLD AND UNDERLINE FONTS INDICATE THE BEST AND THE SECOND BEST PERFORMANCES FOR EACH OBJECT CATEGORY, RESPECTIVELY.

Method	AI	BR	ST	SH	SP	VE	PE	WM
	AP/oLRP	AP/oLRP	AP/oLRP	AP/oLRP	AP/oLRP	AP/oLRP	AP/oLRP	AP/oLRP
anchor-based two-stage:								
TridentNet [10]	9.67/89.84	0.77/98.56	12.28/88.00	17.11/85.00	3.20/97.00	11.87/88.66	3.98/95.80	0.94/98.38
Faster R-CNN [7]	22.71/80.73	3.87/96.23	20.18/81.47	19.02/83.19	8.90/91.50	11.88/88.63	4.49/95.12	0.32/99.08
Cascade R-CNN [24]	**25.57/77.62**	7.47/92.87	23.33/79.07	23.55/79.69	**10.81/89.75**	14.09/86.80	5.34/94.55	0.00/100.00
anchor-based one-stage:								
YOLOv3 [27]	7.14/91.48	2.60/96.72	3.66/95.63	10.69/86.49	0.61/99.26	8.50/89.61	2.13/96.15	0.40/98.80
RetinaNet [26]	0.01/99.88	6.62/93.51	1.84/96.34	20.87/79.40	0.06/99.82	5.67/92.41	1.75/97.04	0.53/99.17
SSD-512 [25]	14.52/86.49	3.13/96.24	10.89/89.40	13.05/87.95	1.92/96.67	7.84/91.22	3.12/96.53	1.48/97.61
anchor-free center-based:								
FoveaBox [34]	13.75/87.26	0.00/100.00	18.51/83.81	17.70/84.88	0.03/99.64	11.42/89.34	3.38/96.19	0.00/100.00
FCOS [33]	14.30/86.46	4.75/94.83	19.77/82.89	22.24/80.97	0.65/98.29	12.51/88.10	3.98/95.62	0.17/99.57
anchor-free keypoint-based:								
RepPoints [31]	2.92/96.18	2.34/97.32	21.37/80.92	**26.40/77.23**	0.00/100.00	15.16/85.90	5.39/94.53	0.00/100.00
Grid R-CNN [29]	22.55/78.59	8.59/91.46	18.93/82.74	21.99/81.21	7.28/92.72	12.94/87.68	4.81/94.99	0.35/99.28
CenterNet [30]	17.43/84.27	9.46/90.61	25.93/75.46	21.86/80.97	6.21/93.42	16.54/82.32	8.12/91.82	1.94/97.73
M-CenterNet	18.59/83.00	**10.58/89.23**	**27.55/74.50**	22.27/79.47	7.53/92.06	**18.60/81.19**	**9.17/90.49**	**2.03/96.73**

and the project for innovative research groups of the Natural Science Foundation of Hubei Province (No. 2018CFA006). In addition, the numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

[1] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3974–3983.

[2] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang, "Mask OBB: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images," *Remote Sensing (Remote Sens.)*, vol. 11, no. 24, p. 2930, 2019.

[3] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing (ISPRS J P&RS)*, vol. 159, pp. 296–307, 2020.

[4] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "$\mathcal{R}^2$-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing (IEEE TGRS)*, vol. 57, no. 8, pp. 5512–5524, 2019.

[5] R. Dong, D. Xu, J. Zhao, L. Jiao, and J. An, "Sig-NMS-based faster r-cnn combining transfer learning for small target detection in vhr optical remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing (IEEE TGRS)*, vol. 57, no. 11, pp. 8534–8545, 2019.

[6] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han, "Scale match for tiny person detection," *CoRR*, vol. abs/1912.10664, 2019.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 91–99.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[9] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.

[10] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks

for object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6054–6063.

[11] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Sod-mtgan: Small object detection via multi-task generative adversarial network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 206–221.

[12] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9725–9734.

[13] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1222–1230.

[14] A. Borji and S. M. Iranmanesh, "Empirical upper-bound in object detection and more," *CoRR*, vol. abs/1911.12451, 2019.

[15] K. Oksuz, B. Can Cam, E. Akbas, and S. Kalkan, "Localization recall precision (lrp): A new performance metric for object detection," in *European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 504–519.

[16] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing (ISPRS J P&RS)*, vol. 98, pp. 119–132, 2014.

[17] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geoscience and Remote Sensing Letters (IEEE GRSL)*, vol. 13, no. 8, pp. 1074–1078, 2016.

[18] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation (JVCIR)*, vol. 34, pp. 187–203, 2016.

[19] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," *CoRR*, vol. abs/1802.07856, 2018.

[20] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu, "Vision meets drones: A challenge," *CoRR*, vol. abs/1804.07437, 2018.

[21] H. Yu, G. Li, W. Zhang, Q. Huang, D. Du, Q. Tian, and N. Sebe, "The unmanned aerial vehicle benchmark: Object detection, tracking and baseline," *International Journal of Computer Vision (IJCV)*, pp. 1–19, 2019.

[22] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," *CoRR*, vol. abs/1912.02424, 2019.

[23] R. Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[24] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6154–6162.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[27] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[28] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 734–750.

[29] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid r-cnn," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7363–7372.

[30] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019.

[31] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9657–9666.

[32] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," *CoRR*, vol. abs/1509.04874, 2015.

[33] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9627–9636.

[34] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Foveabox: Beyond anchor-based object detector," *CoRR*, vol. abs/1904.03797, 2019.

[35] X. Han, Y. Zhong, and L. Zhang, "An efficient and robust integrated geospatial object detection framework for high spatial resolution remote sensing imagery," *Remote Sensing (Remote Sens.)*, vol. 9, no. 7, p. 666, 2017.

[36] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," *CoRR*, vol. abs/1805.09512, 2018.

[37] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for detecting oriented objects in aerial images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2849–2858.

[38] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8232–8241.

[39] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[40] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2403–2412.

[41] K. Chen, J. Wang, J. Pang, and *et al.*, "MMDetection: Open mmlab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.

[44] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision (IJCV)*, vol. 111, no. 1, pp. 98–136, 2015.