Tiny Object Detection in Aerial Images

Jinwang Wang Wen Yang Haowen Guo Ruixiang Zhang Gui-Song Xia

Wuhan University, Wuhan, China

jwwangchn@whu.edu.cn

ICPR2021, Italy, January 2021

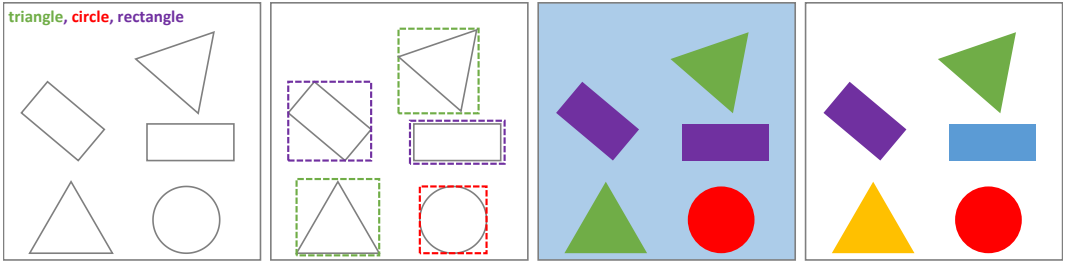
2021/01/13

武漢大學 WUHAN UNIVERSITY

武漢大學 信号处理研究室 dsp.whu.edu.cn

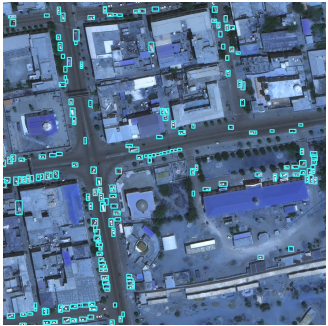
What is object detection?

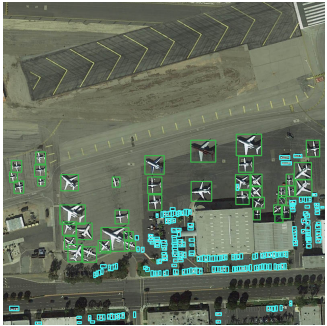
Object detection seeks to locate objects from predefined categories with bounding boxes in an image.



What is tiny object detection in aerial images?

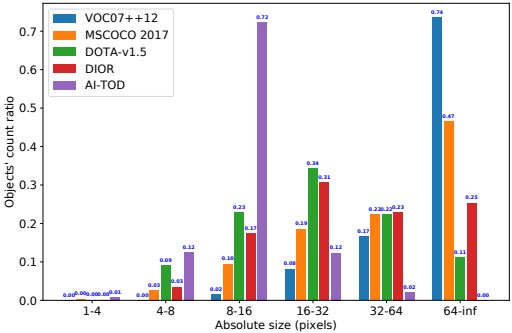
Tiny object is defined as object whose area ratio is less than 0.12% by SPIE. In a typical aerial image with two meters space resolution, ordinary objects like vehicles are usually smaller than 8 pixels.





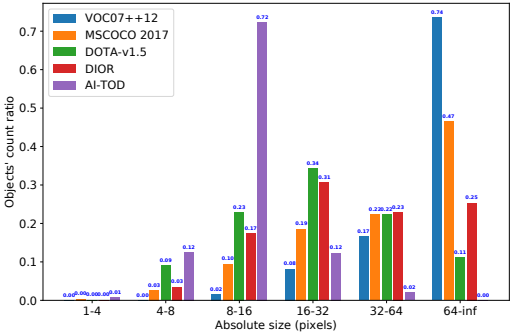
Challenge

There are 67.8% and 79.0% of objects larger than 16 pixels in existing large-scale datasets DOTA and DIOR, respectively. They are not suitable for applications like tiny object detection.



AI-TOD

A new dataset AI-TOD is proposed to handle the scale problem in DOTA and DIOR in this paper. The largest object in AI-TOD is smaller than 64 pixels, 86% of objects in AI-TOD are smaller than 16 pixels, and the mean size of objects in AI-TOD is 12.8 pixels.



Dataset construction process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship and DIOR. We extract images and object instances from the above datasets as follows:

Dataset construction process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship and DIOR. 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.

Dataset construction process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship and DIOR. 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.

(2) Object type. Eight categories *airplane (AI), bridge (BR), storage-tank (ST), ship (SH), swimming-pool (SP), vehicle (VE), person (PE), wind-mill (WM)* are chosen.

Dataset construction process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship and DIOR. 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.

(2) Object type. Eight categories *airplane (AI), bridge (BR), storage-tank (ST), ship (SH), swimming-pool (SP), vehicle (VE), person (PE), wind-mill (WM)* are chosen.

(3) Category conversion. Old categories in the corresponding dataset will be converted to new categories.

Dataset construction process

We build the AI-TOD based on the publicly available large-scale aerial image datasets: DOTA-v1.5, xView, VisDrone2018-Det, Airbus Ship and DIOR. 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.

(2) Object type. Eight categories *airplane (AI), bridge (BR), storage-tank (ST), ship (SH), swimming-pool (SP), vehicle (VE), person (PE), wind-mill (WM)* are chosen.

(3) Category conversion. Old categories in the corresponding dataset will be converted to new categories.

(4) Image selection. The images are chosen by the proportion of tiny objects and the number of large objects in the images.

Dataset construction process

(1) Image size. Original images are divided into 800×800 patches with an overlap of 200 pixels.
(2) Object type. Eight categories *airplane (AI), bridge (BR), storage-tank (ST), ship (SH), swimming-pool (SP), vehicle (VE), person (PE), wind-mill (WM)* are chosen.
(3) Category conversion. Old categories in the corresponding dataset will be converted to new categories.
(4) Image selection. The images are chosen by the proportion of tiny objects and the number of large objects in the images.

After the above processing, 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 with sizes of 800×800 pixels.

Dataset statistics

Number of objects per image set and per class.

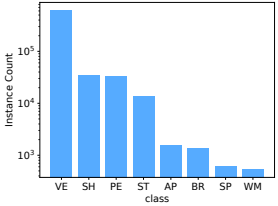
AI-TOD	Train	Validation	Trainval	Test
airplane (AI)	623	170	793	745
bridge (BR)	512	140	652	689
storage-tank (ST)	5,269	2,477	7,746	5,860
ship (SH)	13,539	3,791	17,330	17,633
swimming-pool (SP)	293	34	327	292
vehicle (VE)	248,042	59,904	307,946	306,665
person (PE)	14,126	3,841	17,967	15,443
wind-mill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

Dataset statistics

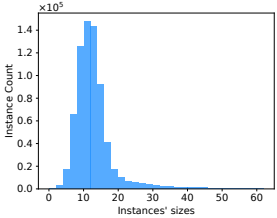
Mean and standard deviation of object scale on different datasets.

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**

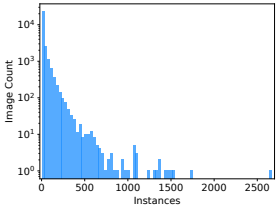
Dataset statistics



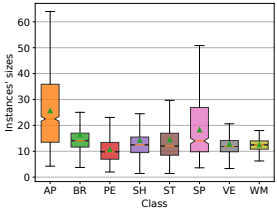
(a) Histogram of the number of instances per class.



(c) Histogram of number of instances' sizes.

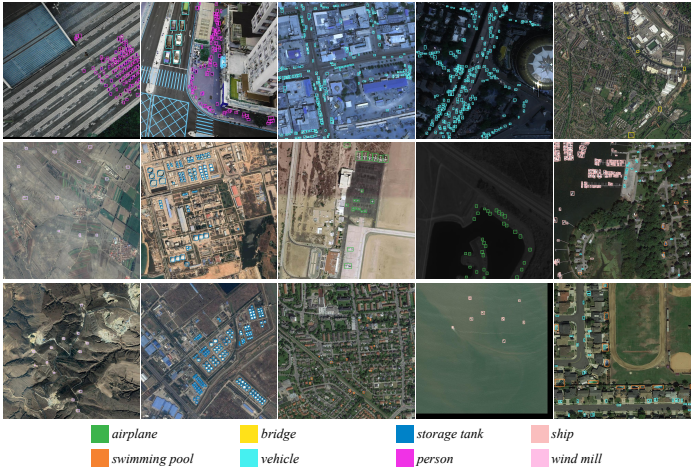


(b) Histogram of number of instances per image.



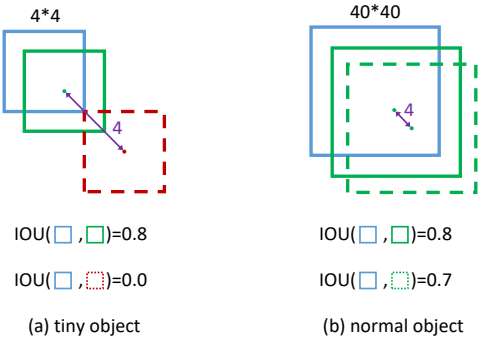
(d) Boxplot depicting the range of sizes for each object category.

Samples


airplane
bridge
storage tank
ship
swimming pool
vehicle
person
wind mill

Sensitivity analysis of IOU

To ensure 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. Thus, localization ability is important for tiny object detector.



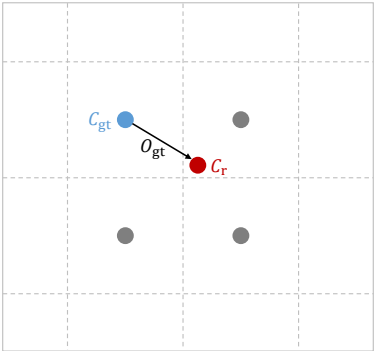
Revisit of CenterNet

CenterNet locates the object by predicting the object center, offset and object size. For obtain high IoU on tiny object, object center and offset are most important.

keypoint heatmap [C] local offset [2] object size [2]

M-CenterNet

Original CenterNet just uses one center point to predict object center and offset.

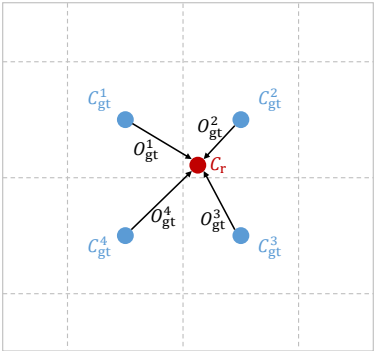


$$C_{\mathrm{gt}} = (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor)$$

$$O_{\mathrm{gt}} = (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor)$$

M-CenterNet

Our proposed Multiple Center Points based Learning Network (M-CenterNet) uses multiple center points to predict object center and offset.



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

$$O_{gt}^1 = (\lfloor cx/s \rfloor, \lfloor cy/s \rfloor),$$
$$O_{gt}^2 = (\lceil cx/s \rceil, \lfloor cy/s \rfloor),$$
$$O_{gt}^3 = (\lfloor cx/s \rfloor, \lceil cy/s \rceil),$$
$$O_{gt}^4 = (\lceil cx/s \rceil, \lceil cy/s \rceil)$$

Performance of twelve detectors on AI-TOD.

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	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	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	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	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	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	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	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	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	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	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	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**

Class-wise object detection results on AI-TOD.

Method	AI	BR	ST	SH	SP	VE	PE	WM
anchor-based two-stage:								
TridentNet	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	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	**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	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	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	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	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	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	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	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	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**

Get AI-TOD Dataset

AI-TOD dataset can be downloaded on the GitHub and Google Drive.

GitHub

https://github.com/jwwangchn/AI-TOD

Google Drive

https://drive.google.com/drive/folders/1mokzFtLCjyqalSEajYTUmyzXvOHAa4WX?usp=sharing

References

G.-S. Xia, X. Bai, et al.
DOTA: A large-scale dataset for object detection in aerial images
CVPR, 2018

K. Li, G. Wan, et al.
Object detection in optical remote sensing images: A survey and a new benchmark,
ISPRS Journal of Photogrammetry and Remote Sensing, 2019

X. Zhou, D. Wang, et al.
Objects as points
arXiv, 2019

H. Yu, G. Li, et al.
The unmanned aerial vehicle benchmark: Object detection, tracking and baseline
IJCV, 2019

K. Chen, J. Wang, et al.
MMDetection: Open mmlab detection toolbox and benchmark
arXiv, 2019

Thanks!