Improving Few-Shot Object Detection through a Performance Analysis on Aerial and Natural Images

Pierre Le Jeune
COSE & L2TI
Université Sorbonne Paris Nord
pierre.le-jeune@cose.fr

Anissa Mokraoui
L2TI
Université Sorbonne Paris Nord
anissa.mokraoui@univ-paris13.fr

***Abstract*—Object detection models require a large amount of annotated data during training, making their deployment for real-world tasks difficult. Few-Shot Object Detection (FSOD) aims to solve this shortcoming by training object detection models from limited data. However, existing methods mostly focus on natural images such as MS COCO and Pascal VOC datasets. In this paper, we study FSOD on aerial images. At first glance, performance seems to decrease compared to natural images with similar datasets (i.e. same number of images and classes). We perform an in-depth analysis to understand the performance discrepancies between natural and aerial images. In the light of this analysis, we propose several improvements to boost the detection quality on aerial images: new data augmentations for object detection, and new support cropping strategies. These modifications increase the mAP by approximately 5% on average.**

***Index Terms*—Few-shot learning, Object detection, Remote sensing images, DOTA, DIOR.**

I. Introduction

Few-shot learning aims to perform a task from only a few annotated examples. Training occurs on a set of base classes for which sufficient data is available, but the overall goal is to deal with novel classes with limited annotations. Few-Shot Object Detection (FSOD) is the intersection of few-shot learning and object detection. It aims ar detecting objects from novel classes in images based on a small set of examples. This is especially important for object detection as it requires a large amount of costly annotated data to achieve satisfactory performance. It is even worse for aerial images which often contain a large number of small objects. FSOD methods attempt to fill this shortcoming by detecting new classes only from several annotated examples (e.g. from 1 to 10, also called shots). An early attempt to tackle this problem is proposed by [1]. Since, several methods have been introduced, but most of them are only benchmarked on natural images datasets such as Pascal VOC [2] and MS COCO [3]. To our knowledge, only two contributions focus on aerial images. The first one [4], extends the feature reweighting method proposed by [1] with multiscale features. The second one [5] develops an adaptive attention mechanism that combines information from the examples (also called support images) with the query image (i.e. the image in which detection is done). We will refer to these methods as FRW and SAA respectively. However, these works do not provide experiments on the same datasets, which prevents direct comparison. As a starting point, we reimplemented both methods (denoted as Few-Shot Baselines) and tested them on one natural image dataset Pascal VOC and two aerial images datasets DOTA [6] and DIOR [7]. Direct performance comparison between different datasets is not possible, therefore, we propose to compare the gaps between the few-shot methods and the regular baseline. The baseline acts as a performance limit, and the distance to this limit shows how well a few-shot method works on a specific dataset. The results can be found in Figure 1. On each dataset, one can observe a slight decrease in performance for base classes, compared to the regular baseline (i.e. without few-shot), and a larger one for novel classes. This behavior is expected in FSOD as only small supervision is available for novel classes, therefore it is harder to detect objects belonging to these classes. However, the performance drop for novel classes is larger for aerial images datasets. We perform a thorough analysis to understand the different behaviors on aerial and natural images.

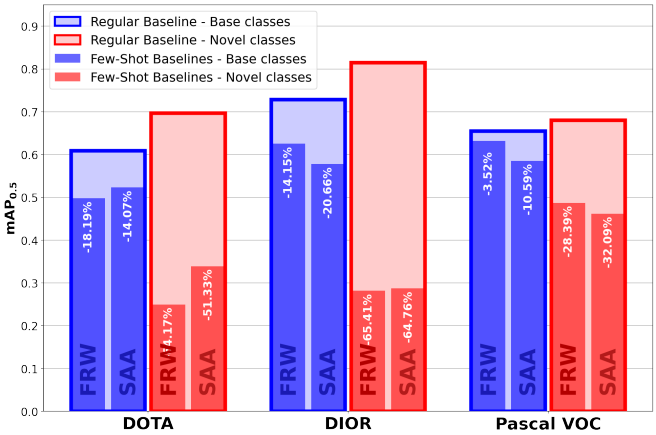

Fig. 1: Performance comparison between Regular Baseline and Few-Shot Baselines, FRW [4] and SAA [5] on three datasets: DOTA, DIOR and Pascal VOC. On aerial image datasets, a large performance gap is observed on novel classes, while this is relatively reduced on Pascal VOC (i.e.natural images).

It appears that the main difference is that objects in aerial images are generally smaller than in natural images. It makes the detection harder, but this is already a known fact: detection models struggle to detect small objects. Solutions have already been proposed to address this issue, such as Feature Pyramidal Network [8] for instance. Here, it is different: there is a larger performance gap between regular baseline and FS baselines

for small objects. We hypothesize that it is harder to extract relevant class information from small objects, and that causes the performance decline. We provide evidence supporting this fact and propose solutions to reduce the performance drop.

The contributions of this work are three-fold:

1) We provide a benchmark of two separate FSOD methods on aerial images datasets DOTA and DIOR.
2) We conduct an in-depth analysis of the performance of FSOD on multiple datasets to better understand the performance gap between aerial and natural images.
3) In the light of this analysis, we improve both methods (FRW and SAA) through a novel information extraction strategy.

II. Performance analysis on natural and aerial images

This work focuses on the performance of FSOD methods on three distinct datasets, DOTA, DIOR and Pascal VOC. Its purpose is to explain why these methods perform better on natural images (i.e. on Pascal VOC). First, we conduct a statistical analysis on the sizes of the objects in each dataset. Then, we look at the performance of FSOD by class for each dataset and correlate this with the dataset analysis.

A. Object size analysis

To begin with, DOTA and DIOR are constituted of aerial images taken from several sources (e.g. Google Earth) at various spatial resolutions, they contain respectively 16 and 20 classes. Pascal VOC is constituted of natural images and contains objects from 20 distinct classes. Table I compares the datasets and highlights statistics on object size. From this, it is obvious that DOTA and DIOR contain far smaller objects than Pascal VOC. In addition, the size variance is greater in DOTA and DIOR (relatively to the average object size). This is detrimental in the few-shot setting as only a small variance can be represented by a few examples. Finally, the surface occupied by objects in Pascal VOC is greater than in the DOTA and DIOR. It leads to a larger object/background imbalance.

Figure 2 shows that DOTA and DIOR both have small classes (i.e. whose median width $\bar{w}$ is below 32 pixels) and large classes (i.e. $\bar{w} > 96$ pixels). On the contrary, Pascal VOC only has medium and large classes. Note that this separation between small/medium/large objects comes from MS COCO and it will be employed extensively in our analysis. This greater size variety in DOTA and DIOR may be detrimental as the network has to deal with both very small and very large objects. In addition, in DOTA and DIOR most classes contain objects of only one size (small, medium, or large), while in Pascal VOC, most spans across two sizes. This certainly forces the network to learn better size robustness through training.

B. Implementation details

To compare the performance of the two selected methods FRW [4] and SAA [5], we implement (from scratch) both of them. The main reason for this is to remove most of the architectural and hyperparameter choices. To do so, we used a recently proposed framework [9] that allows to easily implement a wide variety of FSOD methods. This framework is based on FCOS [10], a one-stage, fully convolutional object detector. Therefore, the two reimplemented methods have the same backbone network and overall architecture. The only difference is the attention mechanism that combines the features from the query image and the support examples. Hence, a fair comparison can be done. For our experiments, we trained FRW and SAA on the three datasets presented above, we refer to these as the few-shot baselines. In addition, a regular FCOS is also trained on the same datasets to compare the performance (i.e. with sufficient data for all classes). All hyperparameters are fixed for these experiments, more details are available in our code[1]. The few-shot baselines are trained following the episodic learning scheme described by [1]. It is separated in two phases: base learning (with base classes only) and fine-tuning (both base and novel classes). Each phase is constituted of multiple episodes which consist in training on a random subset of classes. During an episode, a support set and a query set are sampled containing annotations only for the selected classes. The support set is used as examples by the models to condition the detection on the query images.

	# classes	# instances	Size (in pixels)			Object occupancy
			Mean	Std	Std/Mean	
DOTA	16	190k	33	37	1.12	0.13
DIOR	20	190k	42	58	1.38	0.17
Pascal VOC	20	50k	153	113	0.74	0.40

TABLE I: Object size statistics (in pixel) for DOTA, DIOR and Pascal VOC datasets.

C. Performance comparison on DOTA, DIOR and Pascal VOC

First, with sufficient data for all classes, FCOS performs relatively well on all three datasets. It achieves around 0.65 mAP (with an IoU threshold of 0.5) for DOTA and Pascal VOC while a little under 0.75 mAP for DIOR. With these close results, one could expect a similar behavior when it comes to the FS baselines. Figure 1 actually shows another pattern. On

[1]https://github.com/pierlj/aaf_framework

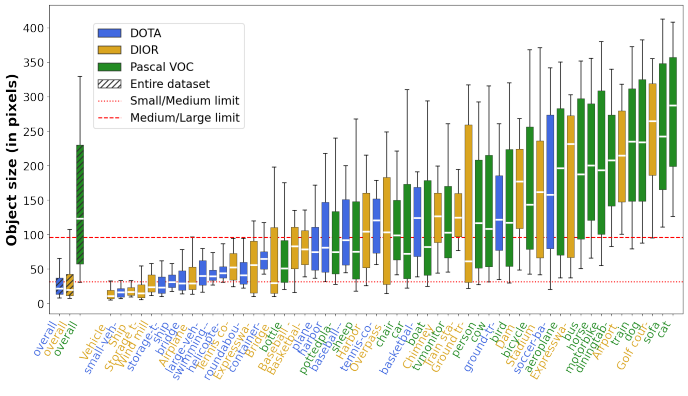

Fig. 2: Box plot of objects size in DOTA, DIOR and Pascal VOC. On the left side, boxes represent the overall size distribution in each dataset. On the right side, the distributions are split by class and ordered by average size.

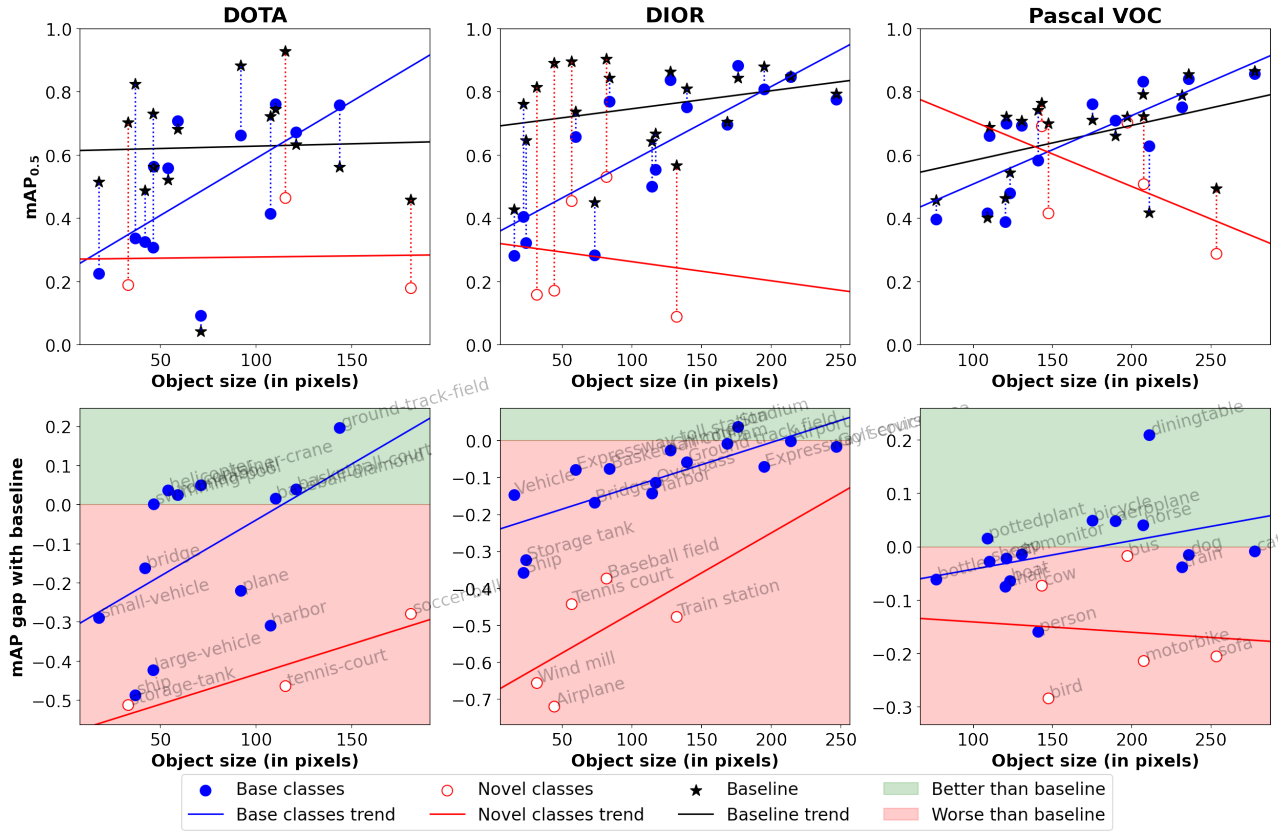

Fig. 3: Performance comparison between FRW baseline (blue and red dots) and regular baseline (black stars) on three different datasets: DOTA, DIOR and Pascal VOC. **(top)** Mean average performance of the two methods plotted per class against average object size. **(bottom)** gap between FRW baseline and regular baseline, per class. Positive values indicate better performance than regular baseline.

the aerial datasets, the FS baselines perform slightly worse than the regular baseline on base classes (i.e. classes with plenty of examples) and poorly on novel classes (about 60% lower). However, on Pascal VOC, the FS baseline is on par with the regular one (even better for FRW) on base classes and 20% lower on novel classes. As the primary goal of FSOD is to get a better performance on novel classes, this makes a large difference. As mentioned in section II-A, objects' sizes vary a lot from one class to another, especially in DIOR and DOTA. Therefore, it may be interesting to look at the performance individually on each class, as depicted in Figure 3. In the top row, the regular baseline and FRW mAP are plotted against the average object size for each class. For FRW, these are split into base and novel classes. From this, it can be observed that generally, the performance increases with the class size, both for the baseline and FRW. Then, performance on novel classes is always under the baseline performance, which is expected given that only a few examples are available for these classes. The smallest base classes show worse performance than the baseline, but for larger ones, FRW outperforms the baseline. To better observe this pattern, the second row of Figure 3 shows the performance gap between the baseline and FRW. A clear trend is visible, having examples available at inference is detrimental for small classes but beneficial for the large one, and the magnitude of this effect increases with the size of the objects. This is also true for novel classes, but as they are more difficult to detect, it remains under the baseline performance. However, this trend is less visible for Pascal VOC. This is due to the lack of small classes (especially for novel classes) required to intensify the trend. If the plots for DOTA and DIOR were restricted to medium and large classes the trends would not be as strong.

To sum up, the smaller objects are harder to detect in general and in the few-shot regime, the information extracted from the examples is detrimental for small classes but beneficial for larger ones. These conclusions are also true for SAA. This highlights a poor choice about the information extraction process in the FS baselines and a lack of robustness in the query/support feature aggregation.

	Base classes				Novel classes			
	Mean	Small	Medium	Large	**Mean**	Small	Medium	Large
Default	0.237	**0.099**	0.261	0.254	0.132	0.034	0.132	0.178
No padding	0.243	0.074	0.281	0.240	0.136	0.034	0.115	0.245
Same size	0.238	0.085	0.271	0.241	**0.153**	0.030	**0.168**	**0.300**
Multi scale	0.231	0.088	0.260	**0.272**	0.145	0.039	0.131	0.255
Reflection	**0.247**	0.086	**0.282**	0.253	0.128	**0.048**	0.139	0.246
Mixed	**0.247**	0.079	0.281	0.247	0.142	0.030	0.124	0.285

TABLE II: Comparison of support extraction strategies on base and novel classes with DOTA dataset and FRW method with 10 shots. The performance is measured as in [3], i.e. mAP is computed with multiple IoU thresholds and separately on objects of different sizes small (S), medium (M) and large (L).

III. Bridging few-shot performance gap on aerial images

A. Improved support feature extraction

As demonstrated in section II-C few-shot performance in detection is closely related to the size of the classes of interest. In particular, the use of examples degrades the quality of detection for small objects. This suggests that the information extracted from the support misguide the model. In our two FS baselines, the support object is cropped out of the example image and zero-padded to fill a 128×128 patch (denoted *extraction method*). This patch is then processed by the backbone network to extract relevant features at multiple scales. Objects larger than 128 pixels are resized to fit in the patch, preserving the aspect ratio. We hypothesize that this is not an optimal strategy as small object crops mostly contain zeros. The features of the objects are diluted with irrelevant information, which could confuse the model during the detection phase. An alternative would be to replace zero padding (denoted **no-padding**) with a larger part of the input image (i.e. simply crop a 128×128 square around the support object), but patches would be dominated by background information, probably harmful as well for detection.

Instead, we propose several novel cropping methods to solve this issue. **Reflection:** instead of zero pad to fill the entire patch, the object is repeated in both directions. Hence, object features are not diluted by zeros nor dominated by background information. **Same-size:** it consists in resizing the support objects to 128×128 patches, preserving the aspect, no matter its original size. It does not change anything for large objects but prevents small objects patches to be dominated by zeros or background. **Multiscale:** each support object is resized at three different scales: small, medium, and large. The three generated patches are 128×128 and contain the same object in different sizes. Each patch is used to compute the features at a different scale. **Mixed:** it uses default extraction strategy for small objects and *same-size* for medium and large ones.

These strategies are compared in Table II. *Same-size* yields noticeable improvements for most classes compared to the default strategy and gives the best overall results. However, it is outperformed by reflection padding when looking only at the performance on base classes. It improves mostly the performance on small and medium objects. For large objects, however, it performs similarly to the default strategy. This is expected as the extracted patches will be almost identical. Using same size support objects simplifies the task for the network as it reduces the intra-class in the support set. It makes it easier to find what are the classes presented in the support set. It is then easier to condition the detection on these classes.

Although it performs better overall, the performance on small objects using the *same-size* strategy is lower compared to the default one. In the case of very small objects, resizing to a 128×128 patch is not optimal as it enlarges a lot the object. The extracted features are highly unlikely to match small objects in the query image. This is the motivation behind the last extraction strategy. However, this is not beneficial, as it performs lower than *same-size* both for base and novel classes. It can be explained by the features discrepancies introduced by having two separate extraction strategies at the same time. Support objects from the same class, with slightly different sizes, could have very different features which could confuse the network during training.

While conceptually simple, it appears that *same-size* is the best extraction strategy. It provides significant performance gains over the default strategy both on base and novel classes.

# Shots		Baseline	+ Flip	+ Color	+ Cutout	+ Crop
1	**Base**	**0.488**	0.458	0.460	0.472	0.457
	Novel	0.062	0.052	0.069	0.064	**0.100**
3	**Base**	**0.511**	0.475	0.470	0.461	0.452
	Novel	0.144	0.186	0.186	0.197	**0.220**
5	**Base**	**0.527**	0.494	0.501	0.503	0.487
	Novel	0.193	0.237	0.251	0.250	**0.259**
10	**Base**	**0.538**	0.508	0.508	0.504	0.503
	Novel	0.286	0.312	0.281	0.341	**0.359**

TABLE III: Cumulative study of the proposed augmentation techniques on DOTA with FRW method. mAP with a 0.5 IoU threshold ($mAP_{0.5}$) is reported for different number of shots.

B. Augmentation for robust query-support matching

Findings from section II-C suggest that the models struggle to match support and query objects when their sizes differ. In addition to the novel support cropping strategy, we propose an augmentation process to improve the robustness of the model and especially the query-support matching. This augmentation process is composed of several random transformations: horizontal and vertical flips, color modifications, cut-out, and random crop-resize. These are all applied to the query images.

It should be noted that these augmentation techniques already exist, however, all are not directly compatible with object detection, especially cut-out and crop-resize as they could produce augmented view with no visible object. The novelty of our work is to adapt cut-out and crop-resize for object detection datasets and apply some augmentation in the support branch to specifically improve the query-support matching in FSOD.

Random cut-out consists in masking a random part of an image. This forces the model to leverage parts of the objects in the decision process and makes it robust to partial occlusions. Generally, a random rectangle is masked out of the image. This works well for classification, but in the context of object detection, a random rectangle could completely mask out some objects (especially small ones). Instead, we choose to apply random cut-out independently on each object in a query image. Thus, multiple rectangles can be sampled for one image, masking only parts of the objects of interest.

Random crop-resize is an augmentation technique that consists in selecting a random rectangular crop in an image and resizing it to the original image size. This changes both the scale and aspect ratio of the objects inside the image. Similar to cut-out, in the context of object detection this must be done carefully to prevent sampling crops without any object inside. We propose to select a random non-empty subset of objects

	DOTA				DIOR					Pascal VOC			
	FRW		SAA		FRW			SAA		FRW		SAA	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	[4]	Baseline	Ours	Baseline	Ours	Baseline	Ours
Base classes	**0.495**	0.485	**0.523**	0.467	**0.625**	0.615	0.540	0.578	**0.618**	**0.647**	0.610	**0.585**	0.531
Novel classes	0.283	**0.371**	0.339	**0.351**	0.282	**0.356**	0.320	0.287	**0.334**	0.522	**0.549**	0.462	**0.488**

TABLE IV: $\text{mAP}_{0.5}$ with 10 shots on three different datasets DOTA, DIOR and Pascal VOC. For each dataset, and each method the table compare the performance with our improvements (augmentation and *same-size* extraction) against the baseline.

$\mathcal{O}$ inside the image, then compute the smallest rectangle r containing all objects in $\mathcal{O}$, and select a random crop between r and the image boundaries. This guarantees having at least one object inside the crop.

Table III shows that all augmentation techniques are beneficial for the performance on novel classes. Altough it produces a slight drop of performance on base classes, the combination improves mAP with 10 shots by 0.07 points on novel classes. This is a significant improvement over the baseline. Similar gains can be observed for the fewer shot settings, which confirms a more resilient query-support matching with augmentation.

IV. Results and experiments

To validate the results of our experiments on DOTA and FRW, we trained both FRW and SAA with the best extraction strategy and the full augmentation pipeline. Each method is trained on DOTA and DIOR to guarantee that the proposed methods improve the performance on aerial images. As a control, we also train on Pascal VOC to verify that our improvements are also superior on natural images. Table IV gathers these results.

It shows that our improvements provide a significant performance gain on novel classes for the two tested methods and on all datasets. Larger increases can be observed for DOTA and DIOR. This was expected as we specifically target aerial images. In addition, our method outperforms significantly existing work on DIOR dataset [4]. On DOTA, such comparison is impossible as no other FSOD method is benchmarked on this dataset. Actually, one contribution [11] that focuses on DOTA exists, but the performance is reported without any fine-tuning on novel classes making the problem even more difficult. Therefore, no fair comparison can be made, and it is not included in Table IV.

V. Conclusion and future work

In a nutshell, we have investigated the performance gap between aerial and natural images for FSOD methods. It seems clear that this gap is mainly due to low performance on small objects. Aerial images contain significantly smaller objects which partly explains the gap. Furthermore, the FSOD performance compared with the baseline performance is correlated with the object size. Using information of the support examples is beneficial for large objects (better performance than the baseline) while detrimental for small objects. This suggests a poor design of the support cropping methods. We partly filled this gap through an augmentation pipeline carefully designed for object detection, and a better support information extraction strategy. Our proposed improvements outperform existing work on DIOR dataset and improve against the baseline on DOTA and Pascal VOC. While this is an encouraging step, there remains a performance gap between FSOD performance on aerial and natural images. Now that a better support information extraction has been found, the query-support matching mechanism must also be designed for small objects.

Acknowledgment

The authors would like to thank COSE for their close collaboration and the funding of this project.

References

[1] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[4] Xiang Li, Jingyu Deng, and Yi Fang. Few-shot object detection on remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 2021.

[5] Zixuan Xiao, Jiahao Qi, Wei Xue, and P. Zhong. Few-shot object detection with self-adaptive attention network for remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:4854–4865, 2021.

[6] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

[7] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020.

[8] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[9] Pierre Le Jeune and Anissa Mokraoui. A unified framework for attention-based few-shot object detection. *arXiv preprint arXiv:2201.02052*, 2022.

[10] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[11] Pierre Le Jeune, Mustapha Lebbah, Anissa Mokraoui, and Hanene Azzag. Experience feedback using representation learning for few-shot object detection on aerial images. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 662–667, 2021.