# Exploring Robust Features for Few-Shot Object Detection in Satellite Imagery

## Supplementary Material

| Type | Category | N=5 | N=10 | N=30 |
|---|---|---|---|---|
| $\mathbf{c}_{base}$ | *car* | 5 | 11 | 32 |
| | *helicopter* | 5 | 10 | 30 |
| | *long-vehicle* | 5 | 9 | 29 |
| | *boat* | 6 | 10 | 33 |
| $\mathbf{c}_{novel}$ | *truck* | 5 | 10 | 30 |
| | *van* | 5 | 11 | 33 |
| | *bus* | 5 | 10 | 31 |
| | *airliner* | 8 | 11 | 36 |
| | *propeller-aircraft* | 5 | 10 | 28 |
| | *trainer-aircraft* | 6 | 10 | 30 |
| | *charted-aircraft* | 5 | 10 | 30 |
| | *fighter-aircraft* | 5 | 11 | 30 |
| | *stair-truck* | 5 | 10 | 34 |
| | *pushback-truck* | 5 | 10 | 30 |

Table 5. Number of instances per class $N$ for each of the used subsets of the SIMD dataset, i.e. $N = \{5, 10, 30\}$. Classes are divided between base classes $\mathbf{c}_{base}$ and novel classes $\mathbf{c}_{novel}$.

## A. Datasets

As mentioned in the main text, satellite images are commonly imbalanced and contain several objects in a single image. This hinders the process of randomly selecting a subset with an equal number of instances per category. Hence, some classes of our subsets have a few examples more or less than $N$. We report the exact number of instances per category in Table 5 and Table 6, for datasets SIMD and DIOR, respectively. In addition, we clarify in the tables which categories belong to novel classes and which ones are base classes. The class *others* of the SIMD dataset is removed from all evaluations, as it is highly underrepresented in the dataset and selecting a subset of approximately $N$ samples per class which includes the *others* category is not trivial. The data splits will be publicly released, containing the images and annotations of each of the subsets.

## B. Implementation details

In this section, we provide further implementation details concerning our evaluation.

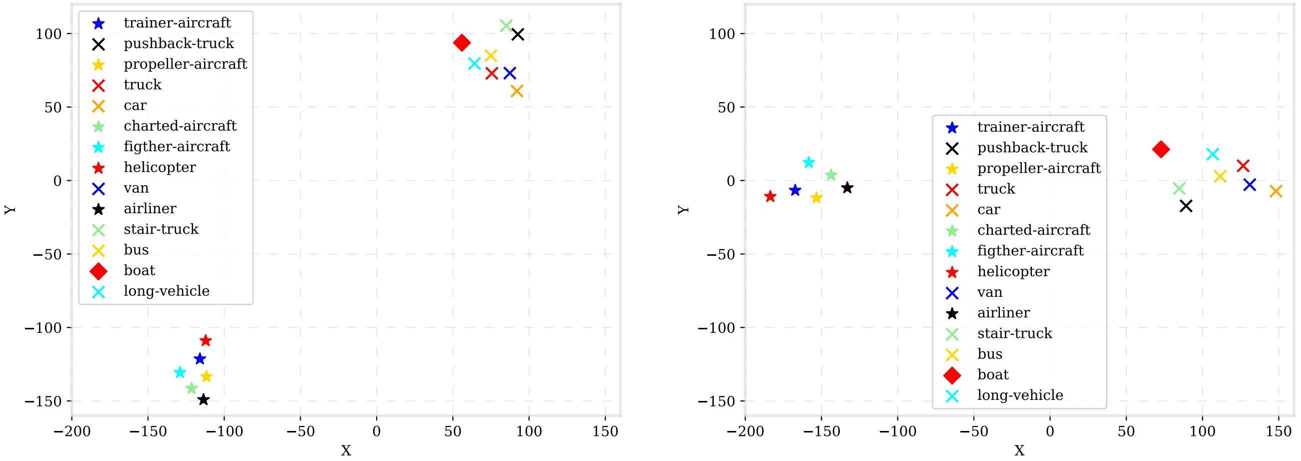
**Ours.** We train our model using Adam optimizer [14] over 200 epochs and a learning rate of $2e^{-4}$. We reduce the learning rate by a factor of 0.1 at epochs 10 and 100. As mentioned in the main text, we apply spatial and radiometric transformations, which involve horizontal and vertical flips with 0.5 probability each, random 90-degree rotation with 0.5 probability, color jitter with $brightness = 0.2$, $contrast = 0.2$, $saturation = 0.2$, and $hue = 0.1$, padding with 0.5 probability, and random crops at a scale of 0.5 to 1. Lastly, crops are resized to $602 \times 602$.

| Type | Category | N=5 | N=10 | N=30 |
|---|---|---|---|---|
| $\mathbf{c}_{base}$ | *ship* | 5 | 10 | 33 |
| | *harbor* | 5 | 11 | 32 |
| | *baseballfield* | 5 | 10 | 30 |
| | *groundtrackfield* | 5 | 10 | 30 |
| | *tenniscourt* | 5 | 10 | 31 |
| | *storagetank* | 5 | 10 | 32 |
| | *airplane* | 5 | 10 | 30 |
| | *basketballcourt* | 5 | 10 | 31 |
| $\mathbf{c}_{novel}$ | *chimney* | 5 | 10 | 30 |
| | *vehicle* | 5 | 11 | 31 |
| | *airport* | 5 | 10 | 30 |
| | *golffield* | 5 | 10 | 30 |
| | *overpass* | 5 | 10 | 30 |
| | *bridge* | 5 | 10 | 30 |
| | *express-toll-station* | 5 | 10 | 30 |
| | *stadium* | 5 | 10 | 30 |
| | *trainstation* | 5 | 10 | 30 |
| | *express-service-area* | 5 | 10 | 30 |
| | *windmill* | 5 | 10 | 31 |
| | *dam* | 5 | 10 | 30 |

Table 6. Number of instances per class $N$ for each of the used subsets of the DIOR dataset, i.e. $N = \{5, 10, 30\}$. Classes are divided between base classes $\mathbf{c}_{base}$ and novel classes $\mathbf{c}_{novel}$.

**YOLO.** We use the source code of YOLOv5 by Ultralytics to pre-train a *yolov5s* model on the entire DOTA dataset. We use 200 epochs with a batch size of 64, using the Ultralytics pre-defined hyper-parameters. The best model was kept and used from there on. Subsequently, we fine-tuned the learned model on the different subsets in two ways. First, we retrain the model on each few-shot subset over 200 epochs with a batch size of 128. Then, we repeat the process but freeze the model's backbone and fine-tune only the heads, thus avoiding overfitting the pre-trained image representations on the small amount of available data. In both setups, we report the results of the best models after the few-shot training.

**DE-ViT.** We use the official implementation of DE-ViT as by their authors. We create masks using the annotations of each subset and generate prototypes with their code. Subsequently, we use their pre-trained model for evaluation of the target datasets with the default values suggested in their publication.



(a) Prototypes without fine-tuning

(b) Prototypes after fine-tuning

Figure 5. T-SNE visualization of the learned prototypes for the SIMD dataset using $N = 10$, before and after fine-tuning. Plane or aircraft types are shown with a star marker, while types of terrestrial vehicles are shown with a cross marker. The *boat* class is shown as a diamond. As depicted, class separation increases after fine-tuning, e.g. *stair-truck* and *pushback-truck* are more separable after training. In addition, each cluster representing a group of transportation exhibits close proximity yet remains distinguishable, whereas the separation between other groups is more pronounced.

**FSRW.** We use the official implementation of the FSRW as by their authors. We perform the model's full training on our end, i.e. training the model in original data and few-shot fine-tuning using the SIMD and DIOR subsets. The hyper-parameters are kept by default as provided by the authors.

## C. Visualization of learned prototypes

Figure 5 provides the T-SNE visualization of the learned prototypes before and after fine-tuning, for the SIMD dataset. We can observe a large separation between groups of classes that represent different types of transportation, i.e. types of planes or aircraft, types of vehicles, and *boat*. Furthermore, the comparison shows that class separability increases after prototype fine-tuning. For example, the class *boat* is quite close to *long-vehicle* and *bus* before fine-tuning, and the distance increases afterward. Similarly, the separability of *stair-truck* and *pushback-truck* increases after learning the prototypes.