**Full name**: Phan Thành Thông

CT UAV Online Assessment

**Question 1**: What are the applicable solutions for few-shot object detection for aerial images? Please provide your solutions (with all scientific references), explaining how it works and why it is feasible.

1. **Meta-Learning–Based Frameworks**
2. **Meta-R-CNN**

**How it works:**

Meta-R-CNN adds meta-learning to both the Region Proposal Network (RPN) and the classification head of Faster R-CNN. In its episodic training process, it learns to:

* Use a class-attention module to highlight features of specific classes while ignoring background noise.
* Train an RPN that generates proposals suited for each new class.
* Use a binary classifier that can quickly adapt to unseen categories using just a few examples.

**Why it is feasible:**

* Better focus on small objects: The class-attention maps help strengthen the signals of tiny objects like vehicles or buildings.
* Quick adaptation: End-to-end meta-training allows the model to recognize new object types (like rare drones or ships) with very few labeled images.

**Reference:** Wu, X., Sahoo, D. and Hoi, S.C.H. (2019) *Meta-RCNN: Meta learning for few-shot object detection*, *Venues*. Available at: https://openreview.net/forum?id=B1xmOgrFPS (Accessed: 24 August 2025).

1. **Meta Faster R-CNN with Attentive Feature Alignment**

**How it works:**

This method improves proposal quality for few-shot classes by:

* Replacing the RPN’s basic object/non-object classifier with a prototype-based matcher (from metric learning), which compares features to class prototypes instead of using a simple linear classifier.
* Using attentive feature alignment to fix spatial mismatches between the proposals and the actual object locations.

**Why it is feasible:**

* Better detection of small, dense objects: Prototype matching helps distinguish tiny, closely packed objects like vehicles or houses.
* Corrects spatial distortions: The alignment step reduces errors from jitter, angle, or perspective shifts that often occur in aerial views.

**Reference:** Han, G. *et al.* (2022) *Meta faster R-CNN: Towards accurate few-shot object detection with attentive feature alignment*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2104.07719 (Accessed: 24 August 2025).

1. **Fine-Tuning–Based Paradigms**
2. **Two-Stage Fine-Tuning Approach (TFA)**

**How it works:**  
TFA trains in two steps:

1. Base Training: A standard Faster R-CNN is trained on abundant base classes (e.g., common aerial objects).
2. Few-Shot Fine-Tuning: The backbone is frozen, and only the box predictor is fine-tuned on a balanced dataset that mixes base and novel classes.

**Why it is feasible:**

* Stable features: Keeping the backbone fixed preserves low-level details (edges, textures) that are vital for detecting small objects.
* Balanced learning: Fine-tuning only the predictor prevents overfitting to new classes while still maintaining accuracy on the original ones.

**Reference:** Wang, X. *et al.* (2020) *Frustratingly simple few-shot object detection*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2003.06957 (Accessed: 24 August 2025).

1. **DeFRCN (Decoupled Faster R-CNN)**

**How it works:**  
DeFRCN extends Faster R-CNN with two key components:

* Gradient Decoupled Layers (GDL): Add learnable transforms and gradient scaling to separate how the backbone interacts with the RPN and with the RCNN head. This lets the RPN and classifier specialize independently.
* Prototypical Calibration Block (PCB): An offline prototype-based classifier that refines RCNN predictions by matching features against class prototypes.

**Why it is feasible:**

* Focused proposal learning: The decoupling allows the RPN to better handle small or difficult objects without being constrained by the classifier.
* Improved classification: Prototype calibration helps distinguish between visually similar aerial objects (e.g., cars vs. trucks, ships vs. boats).

**Reference**: Qiao, L. *et al.* (2021) *DeFRCN: Decoupled faster R-CNN for few-shot object detection*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2108.09017 (Accessed: 24 August 2025).

1. **Low-Rank Adaptation (LoRA) for Cross-Domain Few-Shot Detection**

**How it works:**  
This method applies LoRA to diffusion-based detectors such as DiffusionDet, which are already strong at detecting small objects. LoRA introduces low-rank updates that efficiently adapt pre-trained models to aerial imagery with very few labeled examples. By adding LoRA after fine-tuning on base classes, the model adapts better in extreme few-shot cases (1–5 examples per class). Validated on DOTA and DIOR aerial datasets.

**Why it is feasible:**

* Efficient adaptation: LoRA reduces the number of trainable parameters, lowering the risk of overfitting while saving memory and computation.
* Strong performance in low-shot settings: Maintains accuracy when only a handful of annotated aerial images are available.

**Reference**: Talaoubrid, H. *et al.* (2025) *Analyzing the impact of low-rank adaptation for cross-domain few-shot object detection in aerial images*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2504.06330 (Accessed: 25 August 2025).

1. **Few-Shot Oriented Object Detection with Memorable Contrastive Learning (FOMC)**

**How it works:**  
FOMC tackles two key problems in aerial detection:

* Bounding box misalignment: Replaces standard horizontal boxes with oriented bounding boxes, which better capture objects at arbitrary angles.
* Weak generalization from few samples: Introduces a supervised contrastive learning module with a dynamic memory bank to improve feature discrimination and help recognize unseen categories.

Achieves state-of-the-art performance on DOTA and HRSC2016.

**Why it is feasible:**

* Orientation-aware detection: Handles rotated vehicles, ships, and buildings more accurately than axis-aligned detectors.
* Better generalization: Contrastive learning with memory improves classification even with very limited training data.

**Reference**: Zhou, J. *et al.* (2024) *Few-shot oriented object detection with memorable contrastive learning in remote sensing images*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2403.13375 (Accessed: 25 August 2025).

1. **Exploiting Vision-Language Model Features and Robust Backbone Improvements**

**How it works:**  
This method leverages large pre-trained feature extractors such as DINOv2 to provide stronger general-purpose representations for few-shot detection. In addition, the Region Proposal Network (RPN) is modified to better handle small, dense aerial objects. Combining robust backbones with improved proposals significantly boosts performance in few-shot settings. Demonstrated improvements on DIOR and SIMD satellite imagery datasets.

**Why it is feasible:**

* Generalized features: Pre-trained models like DINOv2 reduce domain gaps between natural images and aerial imagery.
* Better sample efficiency: Stronger features allow the detector to perform well with very few labeled examples.

**Reference**: Bou, X. *et al.* (2024) *Exploring robust features for few-shot object detection in satellite imagery*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2403.05381 (Accessed: 25 August 2025).

**COMPARISON**

| **Method** | **Main Advantage** | **When to Use It** |
| --- | --- | --- |
| **Meta-RCNN** | Learns class-specific proposals and attention maps for stronger few-shot adaptation. | When you have very few novel classes and need quick adaptation (e.g., detecting rare aerial objects like drones or unusual vehicles). |
| **Meta Faster R-CNN w/ Attentive Feature Alignment** | Prototype-based RPN improves recall; alignment fixes spatial mismatches. | When objects are small, crowded, and affected by perspective distortion (e.g., dense urban areas with many cars/buildings). |
| **TFA (Two-Stage Fine-Tuning Approach)** | Stable backbone features and balanced fine-tuning avoid overfitting. | When you have abundant base-class data but few samples for new classes, and want a simple, reliable baseline with low risk of overfitting. |
| **DeFRCN** | Decouples RPN and classifier learning; prototype calibration boosts recognition of similar objects. | When classes are visually similar (e.g., cars vs. trucks, ships vs. boats) and you need robust small-object proposals. |
| **LoRA (Low-Rank Adaptation)** | Efficient adaptation with fewer trainable parameters; strong in 1–5 shot cases. | When computational resources are limited (edge/cloud deployment) or when you must adapt with extremely few samples (1–5 shots). |
| **FOMC (Few-Shot Oriented Obj. Detection)** | Oriented bounding boxes + contrastive learning improve rotation handling and generalization. | When aerial objects are rotated/arbitrarily oriented (e.g., ships in harbors, aircraft on runways, tilted buildings). |
| **Vision-Language & Robust Backbone Features** | Strong pre-trained features reduce domain gaps; improved RPN for small/dense objects. | When you can leverage large pre-trained models (like DINOv2/CLIP) and want generalization across datasets (cross-domain aerial detection). |

1. **DINOv3 x Faster R-CNN (my chosen solution)**

**What is it:**

* DINOv3 is a self-supervised vision transformer (ViT).
* It learns visual features from millions of unlabeled images (without human labels).
* The features are generic and robust, meaning they can be reused for new tasks (like aerial detection).

**How it works:**

* Pretraining: DINOv3 is trained on large image datasets (global features).
* Detection head: Add a detection layer Faster R-CNN head to locate aerial objects (region-level features)

**Why it is feasible:**

* Transfer learning: reuse strong DINOv3 features → fewer labels needed.
* Scalability: works for small objects in cluttered scenes.
* State-of-the-art: Vision Transformers + self-supervised pretraining are proven for few-shot.

**Reference**:

* <https://github.com/facebookresearch/dinov3>
* Siméoni, O. *et al.* (2025) *DINOV3*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2508.10104 (Accessed: 25 August 2025).

**Question 2**: Implement a small demo for your solution in Question 1 (preferably in Pytorch). Please submit your Github link along with a technical report regarding your solution performance and the visualization of the result.

**GITHUB:**