# AeroEyes - Finding and Rescuing With AI-Powered Drones

## I. DESCRIPTION

In emergency and disaster response scenarios, autonomous drones play a crucial role in locating missing persons or critical objects in challenging environments such as flooded zones, forests, or post-storm areas.

This challenge encourages participants to design AI models capable of searching for and localizing a specific object from the drone, based on limited reference images.

Your mission: build a perception system that can determine when and where a given target object appears in drone-captured footage — simulating a real-world search-and-rescue mission.

The qualification round is conducted on pre-recorded drone videos provided by the organizers, simulating real-world aerial search missions:
You are provided with three images of a target object (such as a backpack, person, laptop, bicycle, …) and a drone video scanning an area from above. Your task is to predict the bounding boxes of the object in each detected frame. This is a spatio-temporal localization task that requires recognizing and tracking the target under various scales and viewpoints.

In the final round, the same algorithms will be deployed on real drones for autonomous search in a physical terrain:
The top 5 teams from the qualification phase will compete offline in Ho Chi Minh City and will be responsible for their own participation expenses. Models should also be efficient enough for real-time deployment on Jetson-based drones in the final round.

# II. EVALUATION

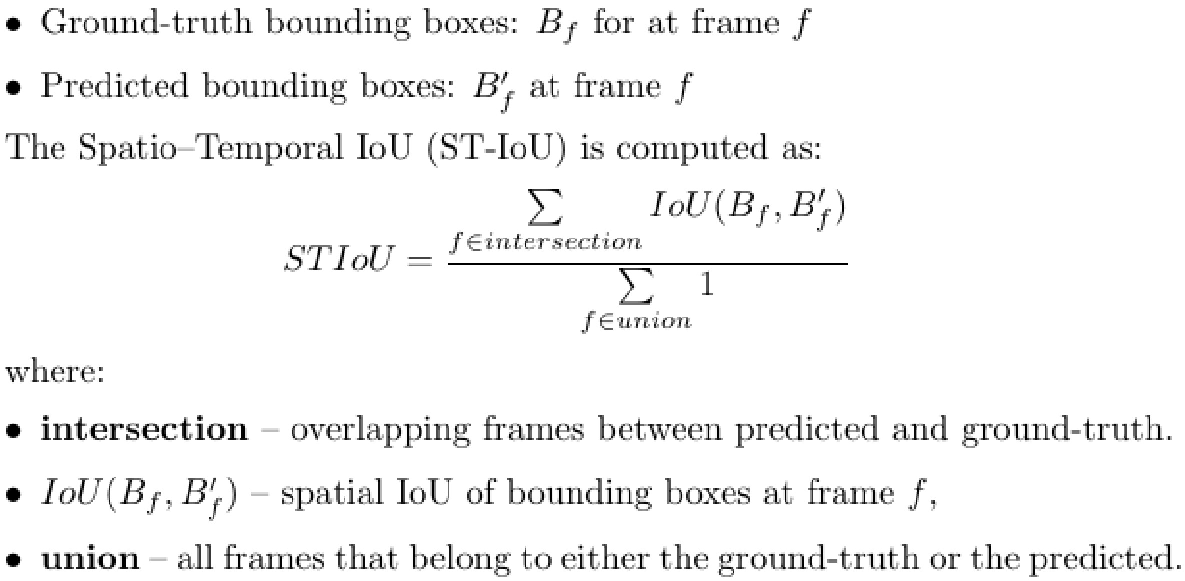
## 1. Qualification round

The evaluation uses a 3-D Spatio–Temporal Intersection-over-Union (ST- IoU) metric that jointly measures when and where the target object is correctly detected in the video. Unlike traditional metrics that treat temporal and spatial accuracy separately, ST-IoU considers them as one continuous space–time volume.

A detection only receives credit if both the timing and the bounding boxes align correctly with the ground truth.

### 1.1 Definition

For each video, let:



- Ground-truth bounding boxes: $B_f$ for at frame $f$
- Predicted bounding boxes: $B'_f$ at frame $f$

The Spatio–Temporal IoU (ST-IoU) is computed as:

$$STIoU = \frac{\sum\limits_{f \in intersection} IoU(B_f, B'_f)}{\sum\limits_{f \in union} 1}$$

where:

- **intersection** – overlapping frames between predicted and ground-truth.
- $IoU(B_f, B'_f)$ – spatial IoU of bounding boxes at frame $f$,
- **union** – all frames that belong to either the ground-truth or the predicted.

### 1.2 Scoring and Aggregation

For each video, the final score is the ST-IoU value between predicted and ground-truth spatio–temporal volumes. The overall leader board score is the mean ST-IoU across all evaluation videos:

$$Final\ Score = \frac{1}{N} \sum_{i=1}^{N} STIoU_{video_i}$$

## 2. Final Round

The top 5 teams from the qualification phase will advance to the on-site final in Ho Chi Minh City. Each team will deploy their model on a real drone equipped with NVIDIA Jetson hardware. The drone must autonomously search for unknown location objects in a physical terrain setup.

Teams will be evaluated based on:

- Detection accuracy (temporal + spatial)
- Search efficiency (time to locate the object)
- Real-time performance (inference speed and stability)

# III. DATA

## 1. Directory Structure

All data are provided in a single folder containing reference images, drone videos, and ground-truth annotations.
Participants are free to create their own training and validation splits for model development.

```
dataset/
├── samples/
│   ├── drone_video_001/
│   │   ├── object_images/
│   │   │   ├── img_1.jpg
│   │   │   ├── img_2.jpg
│   │   │   └── img_3.jpg
│   │   └── drone_video.mp4
│   ├── drone_video_002/
│   │   ├── object_images/
│   │   │   ├── img_1.jpg
│   │   │   ├── img_2.jpg
│   │   │   └── img_3.jpg
│   │   └── drone_video.mp4
│   └── ...
└── annotations/
  └── annotations.json
```

- samples/ — contains all drone video samples and their corresponding reference object images.
- object_images/ — three RGB images of the target object captured from ground-level viewpoints.
- drone_video.mp4 — a 3-5 minute drone-captured video (25 fps) showing the search area.
- annotations/ — JSON file containing ground-truth labels for all samples.

## 2. Ground-Truth Annotation Format

Each record in annotations.json specifies when the target object appears and where it is located within the corresponding video.

```
{
  "video_id": "drone_video_01”,
  "annotations": [
      {
          "bboxes": [
               {"frame": 370, "x1": 422, "y1": 310, "x2": 470, "y2": 355},
               {"frame": 371, "x1": 424, "y1": 312, "x2": 468, "y2": 354},
               {"frame": 372, "x1": 426, "y1": 314, "x2": 469, "y2": 356}
           ]
      }
  ]
}
```

Field Description:
- video_id: unique identifier of the drone video.
- bboxes: list of bounding boxes (x1, y1, x2, y2) per frame (absolute pixel coordinates).
- Each video may contain one or more visible intervals.

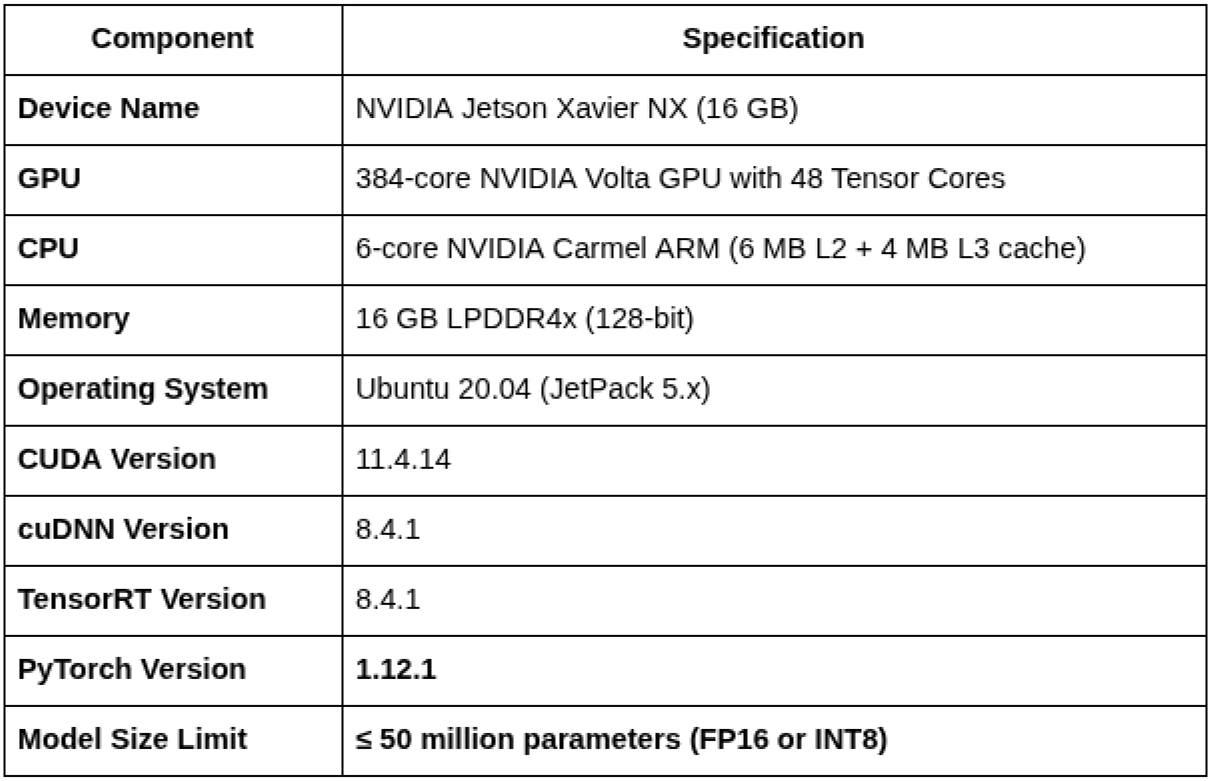
### 3. Expected Submission Format

- Predictions must follow the same schema as the ground-truth annotations.
- Every provided video must appear in the submission file — even if the object is not detected ("detections": []).

```
[
   {
      "video_id": "drone_vid001”,
      "detections": [
           {
              "bboxes": [
                 {"frame": 370, "x1": 422, "y1": 310, "x2": 470, "y2": 355},
                 {"frame": 371, "x1": 424, "y1": 312, "x2": 468, "y2": 354}
               ]
            }
       ]
   },
   {
       "video_id": "drone_video_002",
       "detections": []
   }
]
```

## IV. RULE

- Open-source data/models allowed
- Models must be built and executed directly on the drone. (PyTorch is preferred over TensorRT.). External connectivity (Wi-Fi / cloud inference) is not allowed
- After the competition ends, participants commit not to store any provided data related to the challenge for personal purposes
- The drone runtime environment is configured as follows:



| Component | Specification |
|---|---|
| **Device Name** | NVIDIA Jetson Xavier NX (16 GB) |
| **GPU** | 384-core NVIDIA Volta GPU with 48 Tensor Cores |
| **CPU** | 6-core NVIDIA Carmel ARM (6 MB L2 + 4 MB L3 cache) |
| **Memory** | 16 GB LPDDR4x (128-bit) |
| **Operating System** | Ubuntu 20.04 (JetPack 5.x) |
| **CUDA Version** | 11.4.14 |
| **cuDNN Version** | 8.4.1 |
| **TensorRT Version** | 8.4.1 |
| **PyTorch Version** | **1.12.1** |
| **Model Size Limit** | **≤ 50 million parameters (FP16 or INT8)** |

# V. SUBMISSION GUIDANCE

I. Public Test Round (October 27 – November 20)

- Participating teams build their own solutions, run them on the organizers' public test set, and submit result files (CSV/JSON) via the Submission section on the Zalo AI Challenge portal. Teams are allowed up to 5 submissions per day, and the quota will be reset at the end of each day.Scores and rankings will be displayed on the public leaderboard.
- On November 20, participating teams must submit their complete solutions, including training code, inference code, and data, in the form of a Docker package to the organizers via the "Solution Submission" form on the Zalo AI Challenge portal. Since the Docker file is usually quite large, teams should provide a checksum and a public Google Drive link so that the organizer can download it.

Each team is allowed up to 2 submissions. The organizer will only evaluate the last submission. Submission deadline: 23:59 (GMT+7).

II. Private Test Round (November 25– November 26)

- At 9:00 AM on November 25, the organizer will release the private test set. Participating teams will run their submitted solutions on this set and submit the result files (CSV/JSON) via the Submission section on the Zalo AI Challenge portal, with a deadline of 23:59 (GMT+7) on November 26. The private leaderboard will be displayed in chronological order (based on submission time).

Note: By the end of November or the beginning of December, the top 5 teams from the qualification phase of the AeroEyes - Finding and Rescuing with AI-Powered Drones are expected to compete offline in Ho Chi Minh City. Each team will be responsible for their own participation expenses.

The organizer will evaluate the submitted solutions to verify results, review source code, and detect any cheating. The top 2 teams will be announced on December 11