

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC

**Track 1 - Zalo AI Challenge:
AeroEyes - Finding and Rescuing with
AI-Powered Drones**

MÔN: NHẬP MÔN HỌC SÂU

Giảng viên phụ trách:	Thầy Nguyễn Tiến Huy Thầy Lê Thanh Tùng Thầy Huỳnh Lâm Hải Đăng Thầy Lê Đức Khoan
Nhóm sinh viên thực hiện:	22120133 - Hà Đức Huy 22120099 - Trần Gia Hào 22120101 - Nguyễn Văn Hiến 22120123 - Nguyễn Minh Hưng 22120126 - Nguyễn Tân Hưng

Thành phố Hồ Chí Minh, năm 2025

MỤC LỤC

1	Giới thiệu	2
1.1	Tầm quan trọng của bài toán trong cứu hộ cứu nạn	2
1.2	Định nghĩa Input và Output	2
1.3	Các phương pháp tiếp cận hiện có	3
1.4	Hạn chế của các phương pháp hiện tại	3
1.5	Điểm khác biệt chính trong giải pháp của nhóm	3
2	Các nghiên cứu liên quan	5
2.1	Phát hiện vật thể thời gian thực (Real-time Object Detection)	5
2.2	Học biểu diễn và tìm kiếm dựa trên mẫu (Representation Learning & Few-shot Retrieval)	5
2.3	Theo dõi đối tượng trong video (Object Tracking)	5
3	Phương pháp đề xuất	7
3.1	Tổng quan hệ thống (System Overview)	7
3.2	Thu thập và tiền xử lý dữ liệu	7
3.2.1	Tổ chức dữ liệu đầu vào	7
3.2.2	Trích xuất và chuẩn hóa dữ liệu	8
3.3	Mô hình hóa (Modeling)	9
3.3.1	Kiến trúc mô hình	9
3.3.2	Giai đoạn huấn luyện và thiết lập siêu tham số	9
3.4	Giai đoạn inference và post-processing	11
3.4.1	Khởi tạo mô hình và môi trường suy luận	11
3.4.2	Xử lý ảnh tham chiếu (Reference Embedding)	11
3.4.3	Suy luận YOLO theo luồng video (streaming inference)	11
3.4.4	Chiến lược Scanning và Tracking	12
3.4.5	Hậu xử lý (Post-processing)	12
3.4.6	Xây dựng file kết quả	12
4	Thực nghiệm	14
4.1	Thiết lập dữ liệu và môi trường	14
4.1.1	Phân tích tập dữ liệu	14
4.1.2	Môi trường thực nghiệm	14
4.2	Mô hình cơ sở và so sánh	14
4.3	Thước đo đánh giá (Evaluation Metrics)	14
5	Kết quả	15
5.1	Kết quả định lượng	15
5.2	Kết quả định tính	16
6	Thảo luận	18
6.1	Các yếu tố đóng góp vào việc cải thiện hiệu suất	18
6.2	Ảnh hưởng của chất lượng và sự đa dạng dữ liệu	18
6.3	Khả năng triển khai trong thời gian thực trên drone	18
7	Kết luận	19
	Tài liệu tham khảo	19

1 Giới thiệu

Phần báo cáo này trình bày tổng quan về giải pháp của nhóm cho bài toán AeroEyes tại Zalo AI Challenge 2025. Nội dung bao gồm tầm quan trọng của bài toán, định nghĩa input và output, phân tích các phương pháp hiện có và những đóng góp chính của nhóm trong việc giải quyết các thách thức đặt ra.

Chi tiết cuộc thi: <https://challenge.zalo.ai/portal/aero-eyes>

1.1 Tầm quan trọng của bài toán trong cứu hộ cứu nạn

Trong các kịch bản ứng phó khẩn cấp và cứu hộ thiên tai, thời gian là yếu tố cốt lõi và quan trọng nhất. Các máy bay không người lái (autonomous drones) đóng vai trò quan trọng trong việc tìm kiếm người mất tích hoặc các vật thể quan trọng trong những môi trường phức tạp như khu vực ngập lụt, rừng rậm, hoặc vùng bị tàn phá sau bão.

Việc ứng dụng AI để tự động hóa quá trình tìm kiếm và định vị (localization) từ góc nhìn trên không giúp giảm tải áp lực cho con người, tăng tốc độ xử lý và nâng cao khả năng cứu sống nạn nhân trong những điều kiện mà con người khó tiếp cận. Đây là động lực chính để nhóm phát triển mô hình có khả năng phát hiện đối tượng cần tìm dựa trên số lượng ảnh tham chiếu hạn chế.

1.2 Định nghĩa Input và Output

Bài toán AeroEyes yêu cầu xây dựng một hệ thống xử lý thông tin không gian - thời gian nhằm phát hiện và định vị một đối tượng mục tiêu cụ thể trong dữ liệu video thu thập từ thiết bị bay không người lái (drone). Các mô tả về đầu vào và đầu ra như sau:

- **Input (Đầu vào):**

- **Dữ liệu tham chiếu (Reference Data):** Bao gồm **3 ảnh tham chiếu** của đối tượng mục tiêu. Các ảnh này cung cấp thông tin trực quan đại diện cho đối tượng cần tìm, có thể thuộc nhiều loại khác nhau như con người, ba lô, phương tiện hoặc các vật thể quan tâm khác.
- **Dữ liệu truy vấn (Query Data):** Một **đoạn video** được ghi lại từ drone trong quá trình quét một khu vực tìm kiếm rộng. Video này chứa chuỗi các khung hình theo thời gian, trong đó đối tượng mục tiêu có thể xuất hiện hoặc không xuất hiện tại nhiều thời điểm khác nhau.

- **Output (Đầu ra):**

- **Định vị theo thời gian:** Xác định chính xác các khung hình (frames) trong video mà đối tượng mục tiêu xuất hiện.
- **Định vị theo không gian:** Dự đoán các khung giới hạn (bounding boxes) bao quanh đối tượng trong từng khung hình tương ứng.

Tổng quan, hệ thống cần giải quyết bài toán **định vị không gian-thời gian** (spatio-temporal localization), trong đó không chỉ xác định **đối tượng ở đâu** trong mỗi khung hình mà còn phải xác định **khi nào** đối tượng xuất hiện trong toàn bộ chuỗi video.

Bên cạnh độ chính xác, bài toán còn đặt ra các **ràng buộc về tài nguyên** như là tổng số tham số của mô hình không vượt quá **50 triệu tham số** ($\leq 50M$) hay đáp ứng yêu cầu về phiên bản thư viện sử dụng. Hiệu năng của hệ thống được đánh giá dựa trên chỉ số **Spatio-Temporal Intersection-over-Union (ST-IoU)**, một thước đo phản ánh mức độ chồng lấn giữa dự đoán của mô hình và nhãn chuẩn trong cả không gian lẫn thời gian.

1.3 Các phương pháp tiếp cận hiện có

Hiện nay, bài toán phát hiện vật thể từ drone với dữ liệu tham chiếu ít (Few-shot object detection) thường được giải quyết bằng các hướng tiếp cận sau:

- **Siamese Networks:** Sử dụng kiến trúc song song để trích xuất đặc trưng của ảnh tham chiếu và ảnh query, sau đó so sánh độ tương đồng (similarity matching).
- **Fine-tuning Object Detectors:** Sử dụng các mô hình phát hiện vật thể phổ biến (như YOLO, Faster R-CNN) và tinh chỉnh (fine-tune) nhanh trên tập dữ liệu nhỏ.
- **Vision Transformers (ViT):** Tận dụng cơ chế attention để mô hình hóa mối quan hệ giữa đối tượng tham chiếu và ngữ cảnh toàn cục trong video.

1.4 Hạn chế của các phương pháp hiện tại

Mặc dù đã đạt được một số thành tựu, các phương pháp kể trên vẫn đối mặt với những thách thức lớn trong ngữ cảnh của bài toán AeroEyes:

- **Sự thay đổi về tỷ lệ và góc nhìn:** Đối tượng trong video drone thường rất nhỏ và góc quay thay đổi liên tục so với ảnh tham chiếu tĩnh, khiến các phương pháp so khớp truyền thống kém hiệu quả.
- **Nhiều nền phức tạp:** Môi trường rừng rậm hoặc ngập lụt tạo ra nhiều nhiễu (background clutter), dễ gây nhầm lẫn cho các mô hình dựa trên đặc trưng cục bộ.
- **Ràng buộc về tài nguyên:** Các mô hình Transformer hoặc Two-stage detector thường có kích thước lớn và tốc độ suy luận chậm, khó đáp ứng yêu cầu $\leq 50M$ tham số và triển khai thực tế.

1.5 Điểm khác biệt chính trong giải pháp của nhóm

Khác với các phương pháp tiếp cận truyền thống thường phải đánh đổi giữa *độ chính xác* và *tốc độ suy luận* hoặc phụ thuộc rất lớn vào các kiến trúc Transformer có quy mô lớn và chi phí tính toán cao, giải pháp do nhóm đề xuất tập trung vào việc **tối ưu hóa hiệu năng dưới các ràng buộc tài nguyên nghiêm ngặt**. Cụ thể, nhóm tạo ra sự khác biệt thông qua ba yếu tố cốt lõi:

- **Kiến trúc lai ghép (Hybrid Architecture) để tối ưu hóa tài nguyên:** Thay vì chỉ dựa vào một mô hình duy nhất, hệ thống được thiết kế theo hướng **kết hợp nhiều mô hình chuyên biệt**, mỗi mô hình đảm nhiệm vai trò rõ ràng trong toàn bộ pipeline. Cụ thể, nhóm sử dụng **YOLO11** (phiên bản nano/small) nhằm đảm bảo tốc độ xử lý cao và khả năng hoạt động gần với thời gian thực và **DINOv3** (Self-supervised Learning) để trích xuất đặc trưng ngữ nghĩa sâu. Sự kết hợp này cho phép hệ thống đạt được chất lượng biểu diễn đặc trưng ổn định trong khi vẫn giữ tổng số tham số dưới 50 triệu (50M parameters) theo yêu cầu.
- **Cơ chế xác thực đa tầng (Multi-stage Verification Strategy):** Nhận thấy nhược điểm của các mô hình detection nhẹ là dễ nhận diện nhầm (false positive), nhóm tích hợp một module **xác thực dựa trên màu sắc (Color-based validation)** vào luồng suy luận. Kỹ thuật này đóng vai trò như một bộ lọc cứng, loại bỏ ngay lập tức các ứng viên sai lệch về đặc điểm thị giác cơ bản trước khi xử lý sâu hơn.

- **Tận dụng ngữ cảnh thời gian (Temporal Context Exploitation):** Thay vì xử lý độc lập từng khung hình, hệ thống được mở rộng để tận dụng **thông tin theo chiều thời gian** của dữ liệu video. Nhóm áp dụng cơ chế **bỏ phiếu theo thời gian (Temporal Voting)** kết hợp với kỹ thuật **theo dõi đối tượng liên tục (tracking)** nhằm đảm bảo tính nhất quán của kết quả định vị giữa các khung hình liên tiếp. Cách tiếp cận này giúp ổn định kết quả định vị, giảm thiểu hiện tượng "*nhấp nháy*" (*flickering*) của bounding box và duy trì dấu vết của đối tượng ngay cả trong những khoảng thời gian ngắn mà mô hình tạm thời không đưa ra dự đoán chính xác.

Cách tiếp cận này không chỉ tuân thủ nghiêm ngặt ràng buộc về tài nguyên của cuộc thi mà còn cải thiện chỉ số ST-IoU so với các baseline cơ bản.

2 Các nghiên cứu liên quan

Để giải quyết bài toán AeroEyes với thách thức về định vị không gian - thời gian (Spatio-Temporal Localization) và giới hạn tài nguyên tính toán, nhóm tập trung nghiên cứu ba hướng tiếp cận chính: **Phát hiện vật thể thời gian thực**, **Học biểu diễn thị giác tự giám sát** và **Theo dõi đối tượng**.

2.1 Phát hiện vật thể thời gian thực (Real-time Object Detection)

Trong lĩnh vực thị giác máy tính, các mô hình phát hiện vật thể thường được chia thành hai loại: Mô hình hai giai đoạn (two-stage) và mô hình một giai đoạn (one-stage).

Các mô hình hai giai đoạn như R-CNN hay Faster R-CNN, mặc dù đạt độ chính xác cao nhờ mạng đề xuất vùng (Region Proposal Network), nhưng thường có tốc độ suy luận chậm và chi phí tính toán lớn, không phù hợp cho việc triển khai trên thiết bị bay không người lái (drone) với tài nguyên hạn chế.

Ngược lại, các mô hình một giai đoạn, điển hình là họ mô hình YOLO (You Only Look Once), coi việc phát hiện đối tượng là một bài toán hồi quy duy nhất, dự đoán trực tiếp bounding box và xác suất lớp từ toàn bộ hình ảnh. Sự ra đời của các phiên bản YOLO gần đây (YOLOv8, YOLOv11) đã cải thiện đáng kể sự cân bằng giữa tốc độ và độ chính xác nhờ áp dụng các kiến trúc backbone hiện đại (như CSPNet, ELAN) và các hàm loss tối ưu (CIoU, DFL). Trong bài toán này, nhóm sử dụng kiến trúc YOLOv11 vì khả năng phát hiện tốt các vật thể nhỏ và tuân thủ nghiêm ngặt ràng buộc dưới 50 triệu tham số.

2.2 Học biểu diễn và tìm kiếm dựa trên mẫu (Representation Learning & Few-shot Retrieval)

Bài toán AeroEyes yêu cầu tìm kiếm đối tượng dựa trên một số ít ảnh tham chiếu (reference images), đây là một biến thể của bài toán Few-shot Object Detection. Phương pháp truyền thống thường dựa vào việc so khớp đặc trưng cục bộ (như SIFT, SURF), tuy nhiên các phương pháp này kém hiệu quả khi đối tượng bị thay đổi góc nhìn hoặc điều kiện ánh sáng phức tạp từ drone.

Gần đây, sự phát triển của Vision Transformers (ViT) và học tự giám sát (Self-Supervised Learning - SSL) đã mở ra hướng đi mới. Các mô hình như DINO (Self-distillation with no labels) và các biến thể sau này (DINOv2, DINOv3) đã chứng minh khả năng học được các đặc trưng thị giác mạnh mẽ (visual embeddings) mà không cần nhãn. Các đặc trưng này có tính bất biến cao với các thay đổi về hình thái và màu sắc, cho phép thực hiện việc so khớp ngữ nghĩa (semantic matching) giữa ảnh tham chiếu và đối tượng trong video với độ tin cậy cao hơn nhiều so với các phương pháp CNN truyền thống.

2.3 Theo dõi đối tượng trong video (Object Tracking)

Để đảm bảo tính liên tục theo thời gian, phương pháp *Tracking-by-Detection* hiện là hướng tiếp cận được sử dụng phổ biến trong bài toán theo dõi đối tượng. Phương pháp này bao gồm hai bước chính: (i) Phát hiện đối tượng trong từng khung hình và (ii) Liên kết các kết quả phát hiện theo thời gian để tạo ra các quỹ đạo của đối tượng, dựa trên thông tin vị trí và đặc trưng ngoại hình.

Các thuật toán tiêu biểu như SORT (Simple Online and Realtime Tracking) và DeepSORT sử dụng bộ lọc Kalman (Kalman Filter) để dự đoán trạng thái chuyển động của đối tượng, kết hợp với thuật toán Hungarian (Hungarian Algorithm) nhằm giải quyết bài toán gán nhãn giữa các phát hiện và quỹ đạo hiện có. Trong đó, SORT chủ yếu dựa trên độ chồng lấp không gian (IoU), trong khi DeepSORT bổ sung thêm các đặc trưng ngoại hình được trích xuất bằng mạng học sâu (deep appearance embeddings) để cải thiện khả năng liên kết.

Trong bối cảnh quan sát từ drone, các yếu tố như rung lắc camera, chuyển động nhanh của đối tượng, hiện tượng che khuất (occlusion) và mờ chuyển động (motion blur) có thể dẫn đến việc mất dấu tạm thời. Do đó, việc tích hợp các đặc trưng ngoại hình giàu ngữ nghĩa, đặc biệt là các đặc trưng sâu, đóng vai trò quan trọng trong việc duy trì định danh đối tượng nếu mô hình detection gặp sai lệch trong một số khung hình.

3 Phương pháp đề xuất

Từ những nhận định trên, nhóm thực hiện đề xuất một quy trình xử lý khép kín (end-to-end pipeline). Phần này sẽ trình bày chi tiết về kiến trúc hệ thống, quy trình xử lý dữ liệu, chiến lược huấn luyện mô hình và thuật toán suy luận.

3.1 Tổng quan hệ thống (System Overview)

Hệ thống được thiết kế dựa trên kiến trúc lai ghép (Hybrid Architecture), kết hợp giữa tốc độ của mạng nơ-ron tích chập (CNN) và khả năng hiểu ngữ nghĩa của mạng Transformer. Luồng dữ liệu (Data Flow) của hệ thống diễn ra qua các bước chính sau:

- **Input:** Nhận đầu vào gồm một video quay từ drone và 03 ảnh tham chiếu của đối tượng cần tìm.
- **Feature Extraction:** Ảnh tham chiếu được tách nền và trích xuất đặc trưng sâu (embedding) bằng mô hình DINOv3.
- **Detection Stream:** Video được xử lý theo luồng (stream). Tại mỗi khung hình, mô hình YOLO (đã được tinh chỉnh) sẽ đề xuất các vùng chứa vật thể (Region Proposals).
- **Filtering & Matching:** Các đề xuất từ YOLO được lọc tuân tự qua cơ chế so khớp màu sắc (Color Matching) và so khớp ngữ nghĩa (Semantic Matching với DINOv3).
- **Temporal Tracking:** Các phát hiện hợp lệ được liên kết theo thời gian để tạo thành quỹ đạo ổn định, xử lý các trường hợp bị che khuất tạm thời.
- **Output:** Xuất kết quả dưới dạng danh sách bounding boxes theo định dạng chuẩn của cuộc thi yêu cầu.

3.2 Thu thập và tiền xử lý dữ liệu

Giai đoạn tiền xử lý dữ liệu được nhóm thực hiện nhằm chuyển đổi dữ liệu video thô và tập tin annotations ban đầu về định dạng phù hợp với mô hình YOLO, đồng thời chuẩn hóa cấu trúc để phục vụ huấn luyện.

3.2.1 Tổ chức dữ liệu đầu vào

Dữ liệu gốc được tổ chức theo cấu trúc thư mục gồm:

- Thư mục **train/annotations/** chứa tập tin **annotations.json** lưu trữ toàn bộ thông tin nhãn.
- Thư mục **train/samples/** chứa các video drone, mỗi video nằm trong một thư mục riêng theo **video_id**, với tên tập tin video là **drone_video.mp4**.

Tập tin **annotations.json** chứa danh sách các bản ghi theo video, trong đó mỗi bản ghi bao gồm:

- **video_id:** Mã định danh của video.
- **annotations:** Danh sách các khoảng thời gian có chứa đối tượng cần theo dõi.
- Mỗi khoảng thời gian bao gồm danh sách **bboxes**, trong đó mỗi bounding box được mô tả bởi các trường **x1, y1, x2, y2, frame**.

3.2.2 Trích xuất và chuẩn hóa dữ liệu

Quá trình xử lý bao gồm các bước:

1. Trích xuất khung hình từ video:

Đối với mỗi video, nhóm sử dụng thư viện OpenCV để:

- Mở video bằng `cv2.VideoCapture`.
- Lấy kích thước khung hình gốc gồm chiều rộng và chiều cao.
- Di chuyển đến đúng chỉ số khung hình bằng tham số `CAP_PROP_POS_FRAMES`.
- Đọc và lưu từng khung hình ra file ảnh định dạng `.jpg`.

Mỗi ảnh được đặt tên theo định dạng:

`video_id_frame_XXXXXX.jpg`

trong đó `XXXXXX` là chỉ số khung hình được padding 0 để đảm bảo thứ tự.

2. Chuẩn hóa định dạng bounding box theo YOLO:

Các bounding box ban đầu được lưu dưới dạng tọa độ tuyệt đối theo pixel là (x_1, y_1, x_2, y_2) . Vì thế, nhóm thực hiện chuyển đổi các tọa độ này sang định dạng chuẩn của YOLO là $(x_{center}, y_{center}, w, h)$ và thực hiện chuẩn hóa về khoảng $[0, 1]$ theo công thức:

$$\begin{aligned}x_{center} &= \frac{x_1 + x_2}{2 \cdot W}, & y_{center} &= \frac{y_1 + y_2}{2 \cdot H} \\w &= \frac{x_2 - x_1}{W}, & h &= \frac{y_2 - y_1}{H}\end{aligned}$$

Trong đó W, H là chiều rộng và chiều cao của ảnh gốc. Kết quả được ghi ra tập tin nhãn `.txt` tương ứng với mỗi ảnh, theo định dạng:

`class_id x_center y_center width height`

3. Phân chia tập dữ liệu (Data Splitting):

Thay vì chia theo video, nhóm thực hiện chia ở mức khung hình (frame-level split) để đảm bảo tính đa dạng. Danh sách khung hình có bounding box của mỗi video được:

- Xáo trộn ngẫu nhiên bằng `random.shuffle` với seed cố định là 42.
- Chia thành hai tập theo tỉ lệ:
 - Tập huấn luyện (train): 80%.
 - Tập kiểm thử (validation): 20%.

Cách chia này giúp đảm bảo tính đa dạng của dữ liệu trong cả hai tập nhưng vẫn giữ được tính tái lập (reproducibility) của kết quả.

4. Sinh tập tin cấu hình dataset cho YOLO:

Cuối cùng, nhóm tạo ra tập tin cấu hình `dataset.yaml` phục vụ cho quá trình huấn luyện mô hình YOLO, bao gồm:

- Đường dẫn đến thư mục `train` và `val`.
- Số lượng lớp `nc = 1`.
- Tên lớp đối tượng, ví dụ: `target_object`.

Tập tin này đóng vai trò kết nối giữa dữ liệu đã tiền xử lý và pipeline huấn luyện mô hình phát hiện đối tượng.

5. **Tổ chức cấu trúc thư mục đầu ra:** Sau khi xử lý, dữ liệu được lưu theo đúng cấu trúc chuẩn của YOLO:

- yolo_dataset/images/train/
- yolo_dataset/images/val/
- yolo_dataset/labels/train/
- yolo_dataset/labels/val/
- dataset.yaml phục vụ quá trình train cho YOLO

Mỗi file ảnh luôn có một file nhãn .txt tương ứng cùng tên.

3.3 Mô hình hóa (Modeling)

3.3.1 Kiến trúc mô hình

Nhóm lựa chọn kiến trúc **YOLO11** (phiên bản `yolo11s`) làm backbone cho hệ thống detection. Lý do lựa chọn phiên bản này bao gồm:

- **Cân bằng giữa tốc độ và chính xác:** YOLO11s cung cấp độ chính xác cao hơn phiên bản YOLO11n nhưng vẫn đảm bảo tốc độ suy luận thời gian thực.
- **Kích thước nhỏ gọn:** Tổng số tham số của mô hình nằm trong giới hạn cho phép ($\leq 50M$ parameters).
- **Khả năng phát hiện vật thể nhỏ:** Kiến trúc mới của YOLO11 cải thiện đáng kể khả năng nhận diện các đối tượng kích thước nhỏ - đặc thù của dữ liệu drone.

3.3.2 Giai đoạn huấn luyện và thiết lập siêu tham số

Giai đoạn huấn luyện được thực hiện bằng cách sử dụng thư viện Ultralytics YOLO với kiến trúc mô hình YOLO phiên bản nhẹ (`yolo11s.pt`). Mục tiêu của giai đoạn này là tối ưu trọng số mô hình để học được đặc trưng của đối tượng mục tiêu (target object) từ dữ liệu ảnh được tiền xử lý ở bước trước.

1. Khởi tạo mô hình

Mô hình được khởi tạo từ trọng số tiền huấn luyện (pre-trained weights) qua lệnh:

```
model = YOLO('yolo11s.pt')
```

Việc khởi tạo từ mô hình đã được huấn luyện trước giúp tăng tốc quá trình hội tụ (convergence) và cải thiện hiệu năng trên tập dữ liệu huấn luyện có quy mô hạn chế.

2. Cấu hình dữ liệu huấn luyện

Dữ liệu huấn luyện được cung cấp thông qua tập tin cấu hình `dataset.yaml`, trong đó xác định rõ:

- Đường dẫn đến tập training images và validation images.
- Số lượng lớp đối tượng cần phát hiện.
- Tên lớp đối tượng (`target_object`).

Đường dẫn dataset được chỉ định trực tiếp trong tham số:

```
data='/content/.../yolo_dataset/dataset.yaml'
```

3. Thiết lập siêu tham số huấn luyện

Quá trình huấn luyện được cấu hình với các siêu tham số chính như sau:

- **Số epoch:** epochs = 100, đảm bảo mô hình được huấn luyện đủ số vòng lặp để đạt trạng thái hội tụ.
- **Kích thước ảnh đầu vào:** imgsz = 640, toàn bộ ảnh đầu vào được resize về chuẩn 640×640 .
- **Batch size:** batch = 64, giúp cân bằng giữa tốc độ huấn luyện và mức sử dụng bộ nhớ GPU.
- **Learning rate ban đầu:** lr0 = 0.001, sử dụng làm tốc độ học khởi tạo cho bộ tối ưu.
- **Seed ngẫu nhiên:** seed = 42, đảm bảo tính tái lập (reproducibility) trong việc khởi tạo tham số và chia batch.

4. Kỹ thuật tăng cường dữ liệu (Data Augmentation)

Để tăng khả năng tổng quát hóa của mô hình và giảm hiện tượng overfitting, các kỹ thuật tăng cường dữ liệu được áp dụng trực tiếp trong quá trình huấn luyện:

- **MixUp:** mixup = 0.1, trộn hai ảnh đầu vào cùng nhãn theo hệ số pha trộn nhất định, giúp mô hình học được các đặc trưng mượt hơn giữa các lớp.
- **Xoay ảnh (Rotation):** degrees = 15.0, xoay ngẫu nhiên ảnh trong khoảng $\pm 15^\circ$ để tăng tính đa dạng về góc nhìn.
- **Shear transformation:** shear = 5.0, áp dụng phép biến đổi hình học dạng shear để mô phỏng biến dạng phối cảnh.

5. Quá trình huấn luyện

Quá trình huấn luyện được thực hiện thông qua hàm:

```
model.train(...)
```

Trong quá trình huấn luyện, hệ thống tự động thực hiện các bước sau:

- **Lan truyền xuôi (Forward propagation)** để dự đoán vị trí bounding box, class probabilities và các tham số cần thiết cho bài toán phát hiện đối tượng.
- **Hàm loss** được sử dụng để tối ưu là: **Box Regression Loss** (*box_loss*), **Classification Loss** (*cls_loss*) và **Distribution Focal Loss** (*dfl_loss*).
- **Lan truyền ngược (Backpropagation)** để cập nhật các trọng số của mạng nơ-ron thông qua thuật toán tối ưu (SGD).
- **Dánh giá định kỳ** mô hình trên tập validation sau mỗi epoch, sử dụng các chỉ số như Precision, Recall, mAP@50 và mAP@50–95.

6. Lưu trữ kết quả huấn luyện

Các kết quả huấn luyện được lưu theo tên sau:

```
name = 'drone_training'
```

Kết quả được lưu trữ gồm các thành phần sau:

- Trọng số mô hình theo từng epoch.
- File log quá trình huấn luyện.
- Các chỉ số đánh giá như: Precision, Recall, mAP@0.5, mAP@0.5:0.95.

Các kết quả này được sử dụng cho giai đoạn đánh giá và phân tích hiệu năng mô hình ở các phần tiếp theo của báo cáo.

3.4 Giai đoạn inference và post-processing

Giai đoạn *inference* và *post-processing* được thiết kế như một pipeline hoàn chỉnh nhằm phát hiện và theo dõi đối tượng mục tiêu trong video drone, kết hợp giữa mô hình YOLO, mô hình thị giác tự giám sát DINov3 và các phương pháp hậu xử lý dựa trên đặc trưng màu sắc và chuyển động theo thời gian.

3.4.1 Khởi tạo mô hình và môi trường suy luận

Hệ thống khởi tạo các thành phần chính như sau:

- Nạp mô hình YOLO từ file trọng số đã huấn luyện: `best.pt`.
- Tải mô hình DINov3 từ thư mục cục bộ (`dinov3_local`).
- Khởi tạo phiên làm việc của `rembg` với backend `u2netp` để tách nền đối tượng tham chiếu.
- Tự động lựa chọn thiết bị thực thi (cuda nếu có GPU, ngược lại sử dụng cpu).

Các mô hình được chuyển sang chế độ `eval()` và toàn bộ quá trình suy luận được bao bọc bởi `torch.inference_mode()` nhằm tối ưu bộ nhớ và tốc độ.

3.4.2 Xử lý ảnh tham chiếu (Reference Embedding)

Trước khi quét video, 3 ảnh tham chiếu của đối tượng cần tìm được tải lên từ thư mục và xử lý như sau:

- Chuyển ảnh về định dạng RGB.
- Loại bỏ nền bằng `rembg` (backend `u2netp`) để loại nhiễu background.
- Cắt bỏ vùng nền thừa dựa trên kênh alpha.
- Chuẩn hóa kích thước và phân phối giá trị pixel qua phép biến đổi `torchvision.transforms`
- Trích xuất embedding thị giác bằng mô hình DINov3.

Embedding đại diện cuối cùng được tính bằng trung bình (mean pooling) của các embedding tham chiếu.

3.4.3 Suy luận YOLO theo luồng video (streaming inference)

Mỗi video được xử lý theo chế độ streaming thông qua:

```
model.predict(stream=True)
```

Tại mỗi khung hình:

- Ảnh gốc được lấy từ `results.orig_img`.
- Thực hiện phát hiện bounding box bằng YOLO.
- Loại bỏ các bounding box có kích thước nhỏ hơn ngưỡng quy định (`MIN_BOX_SIZE`).

3.4.4 Chiến lược Scanning và Tracking

Hệ thống hoạt động theo cơ chế chuyển đổi trạng thái (State Machine) giữa hai chế độ:

1. **Chế độ Scanning (Tìm kiếm):** Ở giai đoạn ban đầu, hệ thống duyệt toàn bộ bounding box do YOLO sinh ra và thực hiện xác thực đối tượng bằng hai tiêu chí:

- **Bộ lọc màu (Color Filter):** So sánh độ tương đồng màu sắc giữa vùng crop và ảnh tham chiếu thông qua KMeans clustering trên không gian RGB.
- **Bộ lọc ngữ nghĩa (Semantic Filter):** So sánh cosine similarity giữa embedding DINOv3 của vùng crop và embedding trung bình của ảnh tham chiếu.

Chỉ khi bbox vượt qua cả hai ngưỡng là COLOR_DISTANCE_THRESHOLD và SIM_THRESHOLD thì bbox đó mới được chấp nhận là hợp lệ.

2. **Chế độ Tracking (Theo dõi):** Khi đối tượng được phát hiện liên tiếp trong một số khung hình hợp lệ, hệ thống chuyển sang chế độ Tracking nhằm duy trì định danh và vị trí của đối tượng theo thời gian. Trong chế độ này, hệ thống thực hiện các bước sau:

- Dự đoán vị trí của đối tượng trong khung hình kế tiếp bằng cách ước lượng chuyển động tuyến tính dựa trên vận tốc của các bounding box ở các khung hình trước đó.
- Dánh giá độ phù hợp của bounding box mới thông qua các tiêu chí:
 - Độ chồng lấp không gian (Intersection over Union – IoU);
 - Mức độ dịch chuyển của tâm bounding box (center distance ratio).
- Trong trường hợp bộ phát hiện (YOLO) cung cấp kết quả hợp lệ, hệ thống ưu tiên sử dụng bounding box từ bộ phát hiện thay vì bounding box được suy đoán bởi thuật toán tracking, nhằm hạn chế sai lệch tích lũy theo thời gian.
- Nếu bộ phát hiện không phát hiện được đối tượng trong một số khung hình, hệ thống tạm thời sử dụng bounding box dự đoán từ thuật toán tracking để duy trì quỹ đạo của đối tượng.

Khi số lượng khung hình không có phát hiện vượt quá ngưỡng MAX_BLIND_FRAMES, hệ thống sẽ coi đối tượng đã mất dấu và tự động chuyển về chế độ Scanning.

3.4.5 Hậu xử lý (Post-processing)

Nhằm nâng cao độ ổn định và độ tin cậy của kết quả đầu ra, hệ thống áp dụng một số bước hậu xử lý như sau:

- **Clipping:** Điều chỉnh toạ độ các bounding box để đảm bảo chúng luôn nằm trong phạm vi của khung hình, không vượt quá biên ảnh.
- **Size Filtering:** Loại bỏ các bounding box có kích thước quá nhỏ, vốn thường xuất phát từ nhiễu hoặc các phát hiện không đáng tin cậy.
- **Trajectory Smoothing:** Làm mượt quỹ đạo chuyển động để giảm hiện tượng rung lắc (jitter) của bounding box giữa các khung hình liên tiếp.

3.4.6 Xây dựng file kết quả

Cuối cùng, toàn bộ kết quả được xuất ra tập tin:

submission.json

Cấu trúc tập tin tuân theo chuẩn của ban tổ chức yêu cầu, phục vụ cho việc đánh giá trên hệ thống chấm điểm:

- `video_id`: định danh video.
- `detections`: danh sách các bounding box hợp lệ.
- Mỗi bounding box bao gồm: `frame`, `x1`, `y1`, `x2`, `y2`.

Tóm lại, quy trình này cho phép hệ thống hoạt động gần với thời gian thực (near real-time), đồng thời giảm số lượng false positive thông qua việc kết hợp thông tin không gian, màu sắc và đặc trưng ngữ nghĩa sâu.

4 Thực nghiệm

4.1 Thiết lập dữ liệu và môi trường

4.1.1 Phân tích tập dữ liệu

Dữ liệu thực nghiệm được cung cấp bởi ban tổ chức Zalo AI Challenge 2025, bao gồm các video quay từ drone mô phỏng nhiệm vụ tìm kiếm cứu nạn. Đặc điểm chính của tập dữ liệu này là:

- **Đa dạng môi trường:** Bao gồm các khu vực rừng cây và địa hình phức tạp.
- **Thách thức về tỷ lệ:** Đối tượng cần tìm kiếm thường có kích thước nhỏ so với khung hình tổng thể và thay đổi liên tục do chuyển động của drone.
- **Phân chia dữ liệu:** Như đã trình bày ở phần **Phương pháp đề xuất**, nhóm thực hiện chia tập dữ liệu training/validation theo tỷ lệ 80/20 ở mức khung hình (frame-level) để đảm bảo mô hình học được các đặc trưng đa dạng nhất.

4.1.2 Môi trường thực nghiệm

Toàn bộ thực nghiệm được thực hiện trên môi trường phần cứng với sự hỗ trợ của GPU (CUDA) để đảm bảo tốc độ huấn luyện và suy luận. Các mô hình được cài đặt dựa trên framework PyTorch và thư viện Ultralytics.

4.2 Mô hình cơ sở và so sánh

Để đánh giá hiệu quả của giải pháp, nhóm thực hiện so sánh giữa các phiên bản khác nhau của kiến trúc YOLO thế hệ mới gần đây (v11), cụ thể:

- **Baseline (YOLO11n):** Phiên bản Nano với số lượng tham số cực nhỏ, đóng vai trò làm mốc so sánh về tốc độ và độ chính xác cơ bản.
- **Proposed Method (YOLO11s + Hybrid Pipeline):** Phiên bản Small kết hợp với module hậu xử lý (DINOv3 embedding, lọc màu và tracking) nhằm cải thiện độ chính xác trong các tình huống khó.

Việc so sánh này nhằm kiểm chứng giả thuyết rằng việc tăng nhẹ kích thước mô hình (từ Nano lên Small) kết hợp với các thuật toán lọc nhiễu sẽ mang lại kết quả tốt hơn đáng kể.

4.3 Thước đo đánh giá (Evaluation Metrics)

Chỉ số chính được sử dụng để xếp hạng là **ST-IoU (Spatio-Temporal Intersection-over-Union)**. Khác với IoU truyền thống chỉ tính trên một ảnh 2D, ST-IoU đo lường độ trùng khớp của khối không gian - thời gian (space-time tube) giữa dự đoán và thực tế:

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

Trong đó, một phát hiện **chỉ được tính điểm** nếu khớp cả về mặt thời gian (frame index) và không gian (bounding box). Điểm số cuối cùng là trung bình ST-IoU trên tất cả các video trong tập test.

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

5 Kết quả

5.1 Kết quả định lượng

Bảng 1 trình bày kết quả đánh giá các cấu hình mô hình của nhóm trên bảng xếp hạng (Leaderboard) của cuộc thi Zalo AI Challenge.

Bảng 1: Kết quả thực nghiệm trên Zalo AI Challenge Leaderboard

Mô hình	Cấu hình	Public Score	Private Score
YOLO11n	Base detection	0.46140	-
YOLO11s	Base detection	0.50470	-
YOLO11n	+ Tracking/Filtering	0.53050	0.21530
YOLO11s	+ Tracking/Filtering	0.55460	0.30200

The screenshot shows the Zalo AI Challenge Public Leaderboard. At the top, there are tabs for Overview, Public Leaderboard, Public Submissions, Private Leaderboard, and Private Submissions (which is currently selected). Below this, it displays '2 Private Submissions'. Each submission row contains the file name, upload date, submitter, score, and a link to view members. The first submission is 'of702e19-3f10-4bbd-a20d-193cd1b9815.json' by Hồ Đức Huy with a score of 0.30200. The second submission is '95d121bf-423f-4269-9c28-adea703ad0b.json' by Hồ Đức Huy with a score of 0.21530. A navigation bar at the bottom shows '1 - 2'.

Hình 1: Kết quả nộp trên Private Test

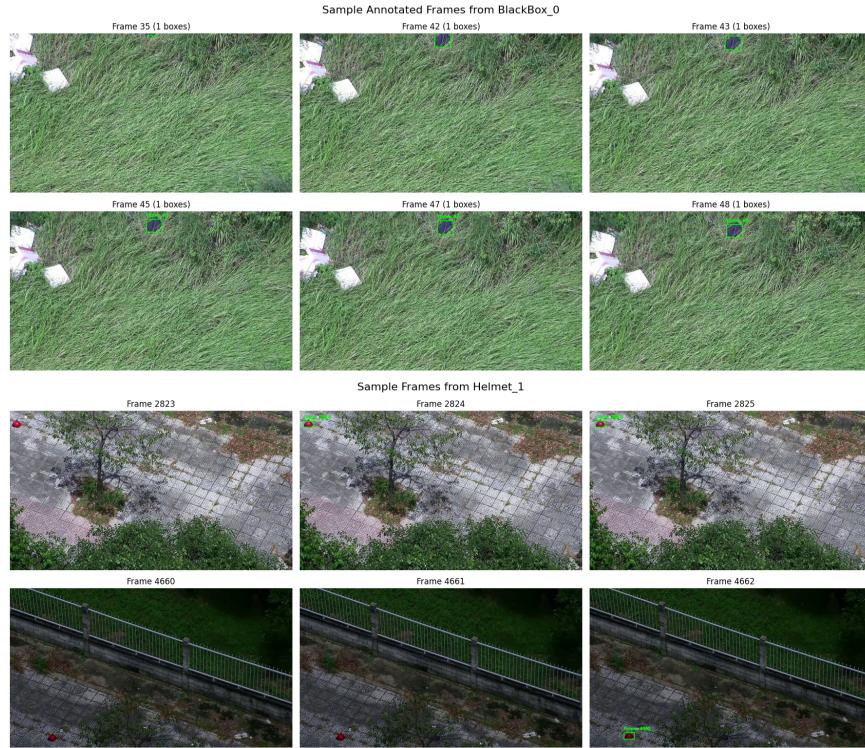
The screenshot shows the Zalo AI Challenge Public Leaderboard. At the top, there are tabs for Overview, Public Leaderboard, Public Submissions, Private Leaderboard, and Private Submissions. The Public Leaderboard tab is selected. It lists various teams and their scores. The team 'Hồ Đức Huy' is ranked 53rd with a score of 0.555. Other teams listed include 'Uia' (0.558), 'Chiron' (0.558), 'TCTT' (0.555), '[HCMUS - FIT] T5H' (0.555), 'Dammonh_AIO' (0.554), 'hehe' (0.542), 'AIQ_Mi 3 miền' (0.538), 'test_submission' (0.537), 'URAx' (0.526), 'BongTraTeam' (0.525), and 'Lotus' (0.519). A navigation bar at the bottom shows '1 - 60'.

Hình 2: Thứ hạng của nhóm trên bảng xếp hạng Public Leaderboard (Nhóm [HCMUS - FIT] T5H)

Nhận xét:

- Trên tập **Public Test**, cấu hình YOLO11s kết hợp với chiến lược Tracking và Filtering đạt điểm số cao nhất (**0.5546**). Kết quả này cho thấy mô hình có khả năng khai thác hiệu quả các đặc trưng dữ liệu có phân phối gần tương đồng với tập huấn luyện.
- Trên tập **Private Test**, điểm số của tất cả các cấu hình đều giảm so với Public Test, phản ánh sự khác biệt về phân phối dữ liệu giữa hai tập đánh giá. Tuy nhiên, việc tích hợp module Tracking và các bước lọc nhiễu hậu xử lý giúp cải thiện tốt hiệu năng. Cụ thể, với cấu hình YOLO11s, điểm số đạt được là **0.3020**, cho thấy pipeline đề xuất giúp mô hình ổn định hơn và cải thiện khả năng tổng quát hóa.
- Với kết quả trên, nhóm cũng đã xếp hạng **53** trên bảng xếp hạng Public của cuộc thi.

5.2 Kết quả định tính



Hình 3: Minh họa kết quả phát hiện đối tượng từ góc nhìn drone. Hộp bao màu xanh thể hiện dự đoán của mô hình khớp với đối tượng mục tiêu.

Quan sát thực tế cho thấy mô hình nhận diện tốt các đối tượng có màu sắc nổi bật và kích thước trung bình. Tuy nhiên, hệ thống gặp khó khăn và hoạt động kém trong một số tình huống phức tạp như sau:

- Đối tượng có kích thước rất nhỏ, chỉ chiếm một số lượng pixel hạn chế khi drone bay ở độ cao lớn.
- Điều kiện ánh sáng bất lợi, đặc biệt là môi trường thiếu sáng hoặc ngược sáng vào thời điểm chiều tối, làm suy giảm độ rõ ràng của các đặc trưng thị giác.

Một vài nguyên nhân dẫn đến những hạn chế nêu trên có thể được lý giải như sau:

- **Thiếu dữ liệu trong điều kiện ánh sáng bất lợi:** Tập dữ liệu huấn luyện chưa bao gồm các khung cảnh ánh sáng yếu hoặc ngược sáng, dẫn đến suy giảm độ chính xác của mô hình YOLO khi đánh giá trên tập *Private Test*.
- **Khả năng nhận diện đối tượng kích thước nhỏ còn hạn chế:** Trong nhiều trường hợp, đối tượng trong video chỉ chiếm vài pixel, vượt quá khả năng nhận dạng của các mô hình nhẹ như YOLO11n/YOLO11s, làm giảm hiệu quả của cả giai đoạn detect và tracking.
- **Dữ liệu huấn luyện chưa được chuẩn hóa tốt:** Dữ liệu được thu thập mang tính tự phát, phân bố không đồng đều giữa các bối cảnh và điều kiện khác nhau, khiến mô hình khó học được các đặc trưng ổn định và có tính khái quát cao.
- **Quy mô dữ liệu còn hạn chế:** Số lượng mẫu huấn luyện chưa tương xứng với độ phức tạp của bài toán, đặc biệt trong bối cảnh đối tượng xuất hiện dưới nhiều điều kiện môi trường và thời tiết khác nhau.

- **Giới hạn về thời gian phát triển:** Thời gian nghiên cứu và thử nghiệm còn hạn chế, khiến nhóm chưa thể đánh giá đầy đủ các kiến trúc mô hình mạnh hơn hoặc áp dụng các chiến lược tăng cường dữ liệu (*data augmentation*) hiệu quả hơn.
- **YOLO đóng vai trò nút thắt của toàn bộ hệ thống:** Hiệu năng của pipeline phụ thuộc trực tiếp vào chất lượng phát hiện ban đầu của mô hình YOLO. Trong các trường hợp mô hình bỏ sót đối tượng, các bước xử lý phía sau như *tracking* hoặc trích xuất đặc trưng (*embedding*) có thể không bù đắp hay khôi phục được thông tin bị mất.

6 Thảo luận

6.1 Các yếu tố đóng góp vào việc cải thiện hiệu suất

Phân tích kết quả thực nghiệm cho thấy hiệu suất của hệ thống không chỉ đến từ mô hình phát hiện đối tượng ban đầu, mà chủ yếu được cải thiện nhờ các thành phần hậu xử lý trong pipeline. Hai yếu tố đóng vai trò quan trọng nhất bao gồm:

- **Cơ chế lọc False Positive:** Việc kết hợp kiểm tra tính nhất quán màu sắc cùng embedding trích xuất từ mô hình DINOv3 hoạt động như một cơ chế xác thực bổ sung, giúp loại bỏ một phần các dự đoán sai của YOLO trong những vùng nền phức tạp. Cách tiếp cận này làm giảm đáng kể nhiễu đầu vào cho các bước xử lý tiếp theo, từ đó cải thiện độ chính xác tổng thể của hệ thống.
- **Tính liên tục theo thời gian (Temporal Consistency):** Chiến lược tracking kết hợp với cơ chế voting theo chuỗi khung hình giúp duy trì sự ổn định của kết quả dự đoán theo thời gian. Cách tiếp cận này đặc biệt hiệu quả trong việc xử lý các *blind frames*, khi đối tượng bị che khuất tạm thời hoặc không được phát hiện ở một số khung hình rời rạc, qua đó hạn chế sự suy giảm của chỉ số ST-IoU.

6.2 Ảnh hưởng của chất lượng và sự đa dạng dữ liệu

Sự chênh lệch đáng kể giữa điểm số trên tập *Public Test* (0.555) và *Private Test* (0.30) cho thấy tác động rõ rệt của hiện tượng **Domain Shift** trong bài toán đặt ra.

- Tập dữ liệu huấn luyện hiện tại chưa bao phủ đầy đủ các điều kiện môi trường đa dạng, đặc biệt là các kịch bản ánh sáng bất lợi như trời tối, ngược sáng hoặc thời tiết xấu (sương mù, mưa).
- Dữ liệu được thu thập mang tính tự phát và phân bố không đồng đều giữa các bối cảnh khác nhau, khiến mô hình có xu hướng *overfitting* vào các đặc trưng phổ biến trong tập Public, nhưng suy giảm khả năng tổng quát hóa khi đánh giá trên tập Private với phân phối dữ liệu khác biệt.

Những quan sát này nhấn mạnh vai trò then chốt của việc xây dựng tập dữ liệu đa dạng và cân bằng trong các bài toán thị giác máy tính triển khai trong môi trường thực tế.

6.3 Khả năng triển khai trong thời gian thực trên drone

Mặc dù vẫn tồn tại những hạn chế về độ chính xác trong các điều kiện môi trường phức tạp, hệ thống của nhóm có thể xem là **phù hợp cho triển khai thời gian thực** trên nền tảng drone, đặc biệt trong các kịch bản yêu cầu phản hồi nhanh như tìm kiếm và cứu hộ.

- **Tối ưu tài nguyên tính toán:** Tổng số tham số của toàn bộ pipeline không vượt quá 50 triệu, đáp ứng tốt các ràng buộc về bộ nhớ và năng lực tính toán của các thiết bị nhúng (*edge devices*) thường được sử dụng trên drone.
- **Độ trễ suy luận thấp:** Việc sử dụng các mô hình lightweight như YOLO11n/YOLO11s, kết hợp với chiến lược suy luận theo luồng (*streaming inference*), giúp đảm bảo tốc độ xử lý ổn định và độ trễ thấp, yếu tố mang tính quyết định đối với các ứng dụng thời gian thực.

7 Kết luận

Trong báo cáo này, nhóm đã trình bày toàn bộ quy trình xây dựng hệ thống nhận diện và truy vết vật thể từ video drone. Hệ thống bao gồm các bước tiền xử lý dữ liệu, xây dựng tập huấn luyện theo định dạng YOLO, huấn luyện mô hình, cũng như thiết kế pipeline suy luận tích hợp nhiều module hậu xử lý như lọc màu, trích xuất embedding bằng DINOv3 và tracking theo thời gian. Toàn bộ giải pháp được thiết kế nhằm đáp ứng các ràng buộc của bài toán, đặc biệt là yêu cầu mô hình nhỏ gọn (dưới 50 triệu tham số, đồng thời tối ưu hiệu năng trong điều kiện dữ liệu hạn chế và môi trường bay phức tạp).

Các phát hiện chính từ quá trình thực nghiệm cho thấy hiệu năng của hệ thống không chỉ phụ thuộc vào mô hình phát hiện đối tượng ban đầu, mà được cải thiện đáng kể nhờ các bước hậu xử lý trong pipeline. Việc kết hợp cơ chế lọc false positive và chiến lược tracking theo thời gian giúp tăng độ ổn định của kết quả dự đoán, đặc biệt trong các trường hợp đối tượng bị che khuất tạm thời hoặc bị bỏ sót ở một số khung hình. Trên bảng xếp hạng của cuộc thi, các cấu hình YOLO11n và YOLO11s lần lượt đạt điểm số 0.53050 và 0.55460 trên Public Leaderboard, cùng với 0.21 và 0.30 trên Private Leaderboard, cho thấy giải pháp đáp ứng tốt yêu cầu của bài toán trong bối cảnh thực nghiệm thực tế.

Tuy nhiên, phương pháp đề xuất vẫn còn một số hạn chế. Đầu tiên, hệ thống gặp khó khăn trong việc phát hiện các đối tượng có kích thước rất nhỏ hoặc xuất hiện trong điều kiện ánh sáng bất lợi, nguyên nhân chủ yếu đến từ sự thiếu đa dạng và chưa được chuẩn hóa của tập dữ liệu huấn luyện. Bên cạnh đó, toàn bộ pipeline vẫn phụ thuộc mạnh vào chất lượng phát hiện ban đầu của mô hình YOLO. Trong các trường hợp mô hình YOLO bỏ sót đối tượng, các bước xử lý phía sau như tracking hay embedding có thể không khôi phục được đầy đủ thông tin bị mất.

Trong tương lai, các hướng cải tiến và mở rộng sẽ tập trung vào việc mở rộng và cân bằng tập dữ liệu nhằm giảm hiện tượng domain shift, nâng cao khả năng phát hiện đối tượng kích thước nhỏ, tích hợp các phương pháp tracking và temporal modeling tiên tiến hơn và giảm phụ thuộc tuyệt đối vào mô hình detection. Những cải tiến này sẽ giúp nâng cao độ chính xác, tăng tính ổn định và giúp hệ thống hoạt động tốt hơn trong môi trường thực tế của drone.

Tài liệu tham khảo

- [1] Zalo AI Challenge, "Track 1: AeroEyes - Finding and Rescuing With AI-Powered Drones". Available at: <https://challenge.zalo.ai/portal/aero-eyes>
- [2] Ultralytics, "Ultralytics YOLO11". Available at: <https://docs.ultralytics.com/vi/models/yolo11/>
- [3] Meta, "DINOv3". Available at: <https://ai.meta.com/dinov3/>
- [4] Daniel Gatis, "rembg". Available at: <https://github.com/danielgatis/rembg>