System Architecture

Pipeline Overview

The complete pipeline consists of five main stages:

Stage	Description
1	Dataset Parsing & Preprocessing
2	Data Augmentation Pipeline
3	YOLOv8 Training
4	Video Inference + ByteTrack
5	ST-IoU Evaluation & Submission

```
Input: Drone Videos + Annotations
    ↓
[STEP 1] Dataset Parsing
    ↓
[STEP 2] Augmentation
    ↓
[STEP 3] Training
    ↓
[STEP 4] Inference
    ↓
[STEP 5] Evaluation
    ↓
Output: submission.json
```

Methodology

Dataset Parsing

Coordinate Transformation

Given bounding box coordinates (x_1, y_1, x_2, y_2) and image dimensions (W, H), we transform to YOLO format:

$$x_{\text{center}} = \frac{x_1 + x_2}{2W}$$

$$y_{\text{center}} = \frac{y_1 + y_2}{2H}$$

$$w = \frac{x_2 - x_1}{W}$$

$$h = \frac{y_2 - y_1}{H}$$

where all values are normalized to the range $[0, 1]$.

Frame Sampling Strategy

To reduce computational cost while maintaining detection coverage:

- Extract all frames with ground truth annotations
- Sample negative frames (no objects) every 50 frames

- Result: approximately 200-300 frames per video

Train/Validation Split
- Split ratio: 80% training, 20% validation
- 14 videos total $\longrightarrow$ 11 training, 3 validation
- Random split (no stratification due to small dataset size)

Data Augmentation

Mosaic Augmentation

Mosaic combines 4 images into a single training sample. For image i in quadrant q, the new bounding box coordinates are:

$$x_{\text{new}} = \frac{x_{\text{old}} \times s_i + \text{offset}_x^q}{W_{\text{mosaic}}}$$

$$y_{\text{new}} = \frac{y_{\text{old}} \times s_i + \text{offset}_y^q}{H_{\text{mosaic}}}$$

where s_i is the scaling factor for image i.

Benefits:
- Increases batch diversity
- Improves small object detection
- Simulates multiple objects in scene

Random Scaling

Scale range: $s \in [0.5, 1.5]$

For image I with dimensions (W, H):

$$W_{\text{new}} = W \times s$$

$$H_{\text{new}} = H \times s$$

Corresponding bounding box transformation:

$$x_{\text{center}}^{\text{new}} = \frac{x_{\text{center}} \times s}{W_{\text{new}}}$$

$$y_{\text{center}}^{\text{new}} = \frac{y_{\text{center}} \times s}{H_{\text{new}}}$$

Rotation

Rotation angle: $\theta \in [-10°, +10°]$

Why limited rotation?
- Aerial perspective constraint: sky always up, ground always down
- Large rotations break semantic meaning
- Small rotations simulate drone tilt

Rotation matrix:

$$\boldsymbol{R}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

HSV Color Jitter
Transformation parameters:

$$H_{\text{new}} = H + \delta_h \times 180° \quad \text{where } \delta_h \in [-0.015, 0.015]$$

$$S_{\text{new}} = S \times (1 + \delta_s) \quad \text{where } \delta_s \in [-0.7, 0.7]$$

$$V_{\text{new}} = V \times (1 + \delta_v) \quad \text{where } \delta_v \in [-0.4, 0.4]$$

Purpose: Simulate varying lighting conditions (dawn, noon, dusk, cloudy)

MixUp Augmentation
MixUp creates soft samples by linear interpolation:

$$x_{\text{mixed}} = \lambda x_i + (1 - \lambda) x_j$$

$$y_{\text{mixed}} = \{y_i \cup y_j\}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$ with $\alpha = 0.2$

YOLOv8 Training

Model Architecture
Backbone: CSPDarknet
- Cross Stage Partial connections
- Reduces computational redundancy
- Maintains gradient flow

Neck: Path Aggregation Network (PAN)
- Bottom-up and top-down feature fusion
- Multi-scale feature extraction

Head: Decoupled Detection Head
- Separate branches for classification and localization
- Anchor-free detection

Loss Function
The total loss is a weighted combination of three components:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}} \mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{dfl}} \mathcal{L}_{\text{dfl}}$$

where $\lambda_{\text{box}} = 7.5$, $\lambda_{\text{cls}} = 0.5$, $\lambda_{\text{dfl}} = 1.5$

Box Loss (Complete IoU)

$$\mathcal{L}_{\text{box}} = 1 - \text{CIoU}\left(B_{\text{pred}}, B_{\text{gt}}\right)$$

$$\text{CIoU} = \text{IoU} - \rho^2 \frac{b, b_{\text{gt}}}{c^2} - \alpha v$$

where:
- $\text{IoU} = \left|B_{\text{pred}} \cap B_{\text{gt}}\right| \left| B_{\text{pred}} \cup B_{\text{gt}}\right|$
- ρ = Euclidean distance between centers
- c = diagonal length of smallest enclosing box
- $v = \left(\frac{4}{\pi^2}\right)\left(\arctan\left(\frac{w_{\text{gt}}}{h_{\text{gt}}}\right) - \arctan\left(\frac{w}{h}\right)\right)^2$
- $\alpha = \frac{v}{1 - \text{IoU} + v}$

Classification Loss
Binary cross-entropy for each class:

$$\mathcal{L}_{\text{cls}} = -\sum_i [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

Distribution Focal Loss
For bounding box coordinate regression:

$$\mathcal{L}_{\text{dfl}} = -\sum \left[\left(y_{\text{left}} \log(p_{\text{left}}) + y_{\text{right}} \log(p_{\text{right}}) \right) \right]$$

Optimization
Optimizer: Stochastic Gradient Descent (SGD)

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t) + \mu v_t$$

where:
- $\eta = 0.01$ (learning rate)
- $\mu = 0.937$ (momentum)
- weight decay $= 0.0005$

Learning Rate Schedule: Cosine Annealing

$$\eta_t = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \frac{1 + \cos(\pi \frac{t}{T})}{2}$$

where $\eta_{\max} = 0.01$, $\eta_{\min} = 0.0001$, $T = 100$ epochs

Warmup: First 3 epochs

$$\eta_t = \eta_{\text{base}} \times \left(\frac{t}{T_{\text{warmup}}} \right) \quad \text{for } t \in [0, 3]$$

Evaluation Metrics
Mean Average Precision at IoU=0.5:

$$\text{AP} = \int_0^1 P(R) \, \mathrm{d}R$$

$$\text{mAP@0.5} = \frac{1}{N} \sum_i \text{AP}_i$$

where:
- $P = \frac{\text{TP}}{\text{TP + FP}}$ (Precision)
- $R = \frac{\text{TP}}{\text{TP + FN}}$ (Recall)
- N = number of classes

Mean Average Precision at IoU=[0.5:0.95]:

$$\text{mAP@}[0.5 : 0.95] = \frac{1}{10} \sum_{\text{IoU}=0.5}^{0.95} \text{mAP@IoU}$$

Video Inference with ByteTrack

Detection Phase

For each frame f_t, apply YOLOv8:

$$\mathcal{D}_t = \text{YOLOv8}(f_t) = \{(b_i, c_i, s_i) \mid i = 1...N\}$$

where:

- b_i = bounding box $[x_1, y_1, x_2, y_2]$
- c_i = class ID
- s_i = confidence score

ByteTrack Algorithm

Step 1: High-Confidence Track Association

For detections with $s_i > \tau_{\text{high}}$ ($\tau_{\text{high}} = 0.4$):

$$\text{IoU}(b_{\text{det}}, b_{\text{track}}) = \frac{\text{Area}(b_{\text{det}} \cap b_{\text{track}})}{\text{Area}(b_{\text{det}} \cup b_{\text{track}})}$$

Match if $\text{IoU}(b_{\text{det}}, b_{\text{track}}) > \tau_{\text{match}}$ ($\tau_{\text{match}} = 0.8$)

Step 2: Low-Confidence Re-identification

For detections with $\tau_{\text{low}} < s_i < \tau_{\text{high}}$ ($\tau_{\text{low}} = 0.1$):

Match with lost tracks if $\text{IoU}(b_{\text{det}}, b_{\text{lost}}) > \tau_{\text{rematch}}$ ($\tau_{\text{rematch}} = 0.5$)

Step 3: Track Management

$$\text{lost\_frames}_i = \begin{cases} 0 & \text{if matched at frame } t \\ \text{lost\_frames}_i + 1 & \text{if not matched} \end{cases}$$

Remove track if $\text{lost\_frames}_i > \text{buffer\_size}$ ($\text{buffer\_size} = 30$)

Temporal Smoothing

Moving Average Filter:

For track trajectory $\mathcal{T} = \{b_1, b_2, ..., b_t\}$:

$$b_{\text{smooth}(t)} = \frac{1}{w} \sum_{i=t-w+1}^{t} b_i$$

where $w = 5$ (window size)

Component-wise smoothing:

$$x_1^{\text{smooth}} = \frac{1}{5} \sum_{i=t-4}^{t} x_1^i$$

Benefits:

- Reduces jitter in bounding box coordinates
- Stabilizes tracking visualization
- Smooths motion trajectory

Spatio-Temporal IoU Metric

Definition

Given predictions $\mathcal{P}$ and ground truth $\mathcal{G}$ for video $\mathcal{V}$:

$$\text{ST-IoU}(\mathcal{P}, \mathcal{G}) = \frac{\sum_{f \in \mathcal{F}_{\text{intersect}}} \text{IoU}\left(B_f^{\mathcal{P}}, B_f^{\mathcal{G}}\right)}{|\mathcal{F}_{\text{union}}|}$$

where:

- $\mathcal{F}_{\text{intersect}}$ = frames with both prediction and ground truth
- $\mathcal{F}_{\text{union}}$ = frames with prediction OR ground truth
- $B_f^{\mathcal{P}}$ = predicted bbox at frame f
- $B_f^{\mathcal{G}}$ = ground truth bbox at frame f

Bounding Box Matching

For multiple objects in frame f, use greedy matching:

1. Compute IoU matrix $\boldsymbol{M}[i, j]$ for all prediction-GT pairs
2. While unmatched pairs exist:
   - Find $\max \text{IoU}$ in $\boldsymbol{M}$
   - If $\max \text{IoU} > \text{threshold}$:
     - Match pair (i^*, j^*)
     - Remove row i^* and column j^*
   - Else: break

Example Calculation

Given:

- Ground Truth: frames $\{100, 101, 102\}$
- Prediction: frames $\{101, 102, 103\}$

Frame 101:

- GT: $[50, 50, 100, 100]$
- Pred: $[52, 48, 98, 102]$
- Intersection $= (98 - 52) \times (100 - 48) = 2392$
- Union $= 2500 + 2496 - 2392 = 2604$
- $\text{IoU}_{101} = \frac{2392}{2604} = 0.919$

Frame 102:

- $\text{IoU}_{102} = 0.890$

Final ST-IoU:

$$\mathcal{F}_{\text{intersect}} = \{101, 102\}$$

$$\mathcal{F}_{\text{union}} = \{100, 101, 102, 103\}$$

$$\text{ST-IoU} = \frac{0.919 + 0.890}{4} = 0.452$$

TensorRT Optimization

Model Conversion Pipeline

```
PyTorch (.pt)
    ↓ [export to ONNX]
ONNX (.onnx)
    ↓ [TensorRT builder]
TensorRT Engine (.engine)
```