Multi person 3D pose estimation from monocular camera: Application to the identification and the tracking of subgroups

Dam Ngoc Khanh

Advisor: Pr. Vincent BARRA

Academic advisor: Dr. Hervé KERIVIN

ISIMA - Université Clermont Auvergne

Sep 04, 2025

Outline

Introduction to the topic

- Object detection and tracking are important problem in the fields of Data Science and Computer Vision.
- Apply to the identification and the tracking of subgroups from monocular video, especially students in school yard.
- Monocular cameras are cost-effective but lack of the depth.

Research Objectives and Tasks

Research Objectives: Build a solution to detect and track groups of people, mainly videos of students in the school yard.

Main tasks:

- Explore object detection and multi-object tracking algorithms.
- Develop a grouping algorithm and tracking groups.
- Experiment and evaluate to give improvements and refines the system.

Proposed Approach

Three main modules in the pipeline:

- Human detector: Human detection with bounding box coordinates.
- Human tracking: Track people and assign unique IDs to each person.
- Grouping algorithm: Identify groups and maintain monitoring of the group's status.

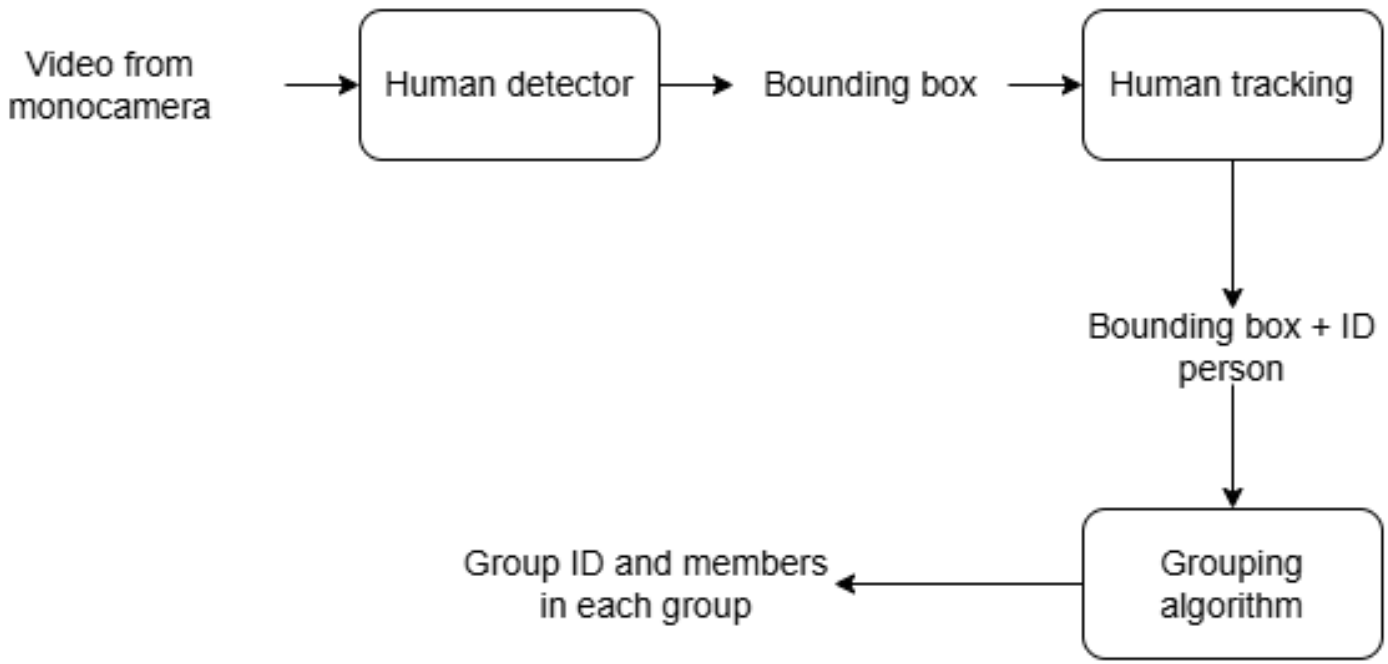


Figure: Pipeline overview of group detection and tracking system

Pipeline overview

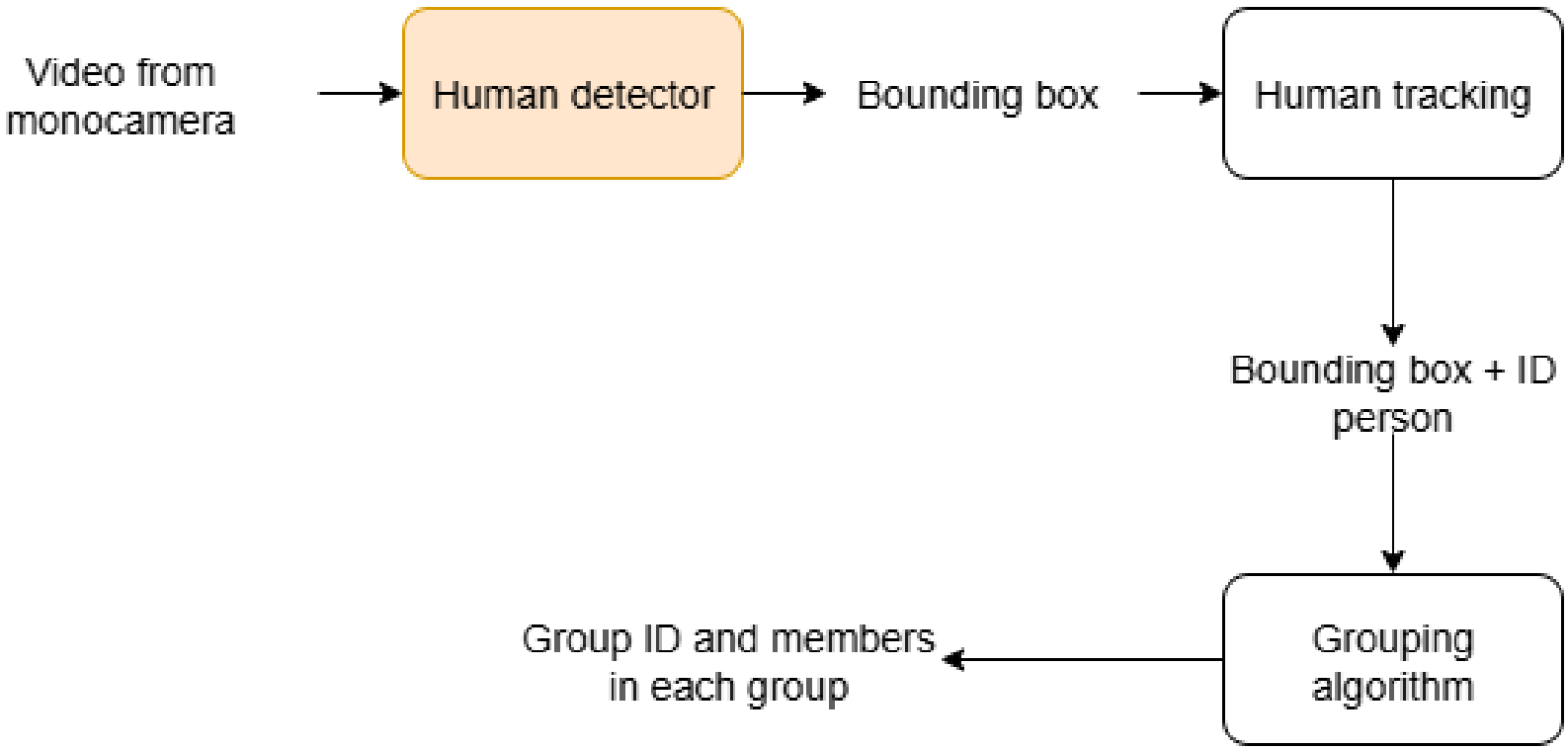


Figure: Pipeline overview of group detection and tracking system

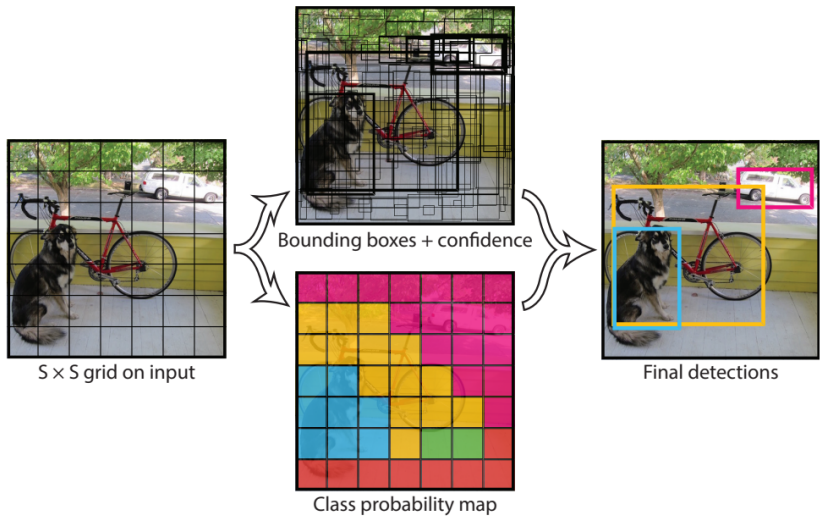
Human detector

- One-stage detector:
  - Perform detection in a single step by directly predicting class probabilities and bounding boxes.
  - Designed for speed and real-time applications.
  - Examples: YOLO, SSD, RetinaNet.
- Two-stage detector:
  - Work in two steps: first generate region proposals, then classify and refine them.
  - Achieve higher accuracy, but are generally slower.
  - Examples: R-CNN, Faster R-CNN, Mask R-CNN.

Human detector: One-stage

YOLO (You Only Look Once) is a pioneering object detection model that processes images in a single pass through the neural network, first introduced by Joseph Redmon et al. in 2015.

- One-stage detectors apply a single neural network to the full image.
- The image is divided into a grid of cells.
- The network generates anchor boxes of varying scales and sizes and predicts class probabilities.

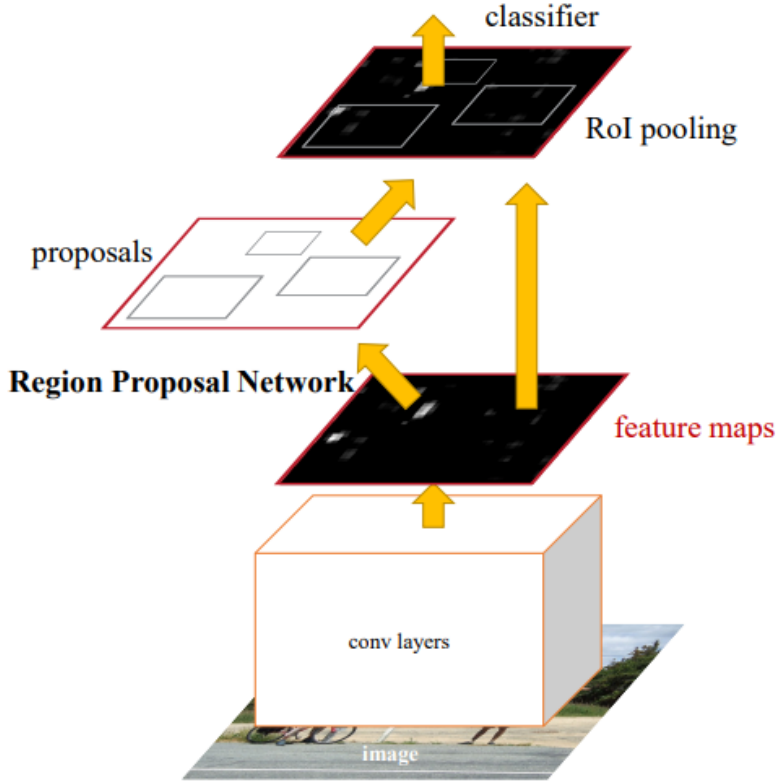


$\Longrightarrow$ Makes significantly faster than traditional methods, making it well-suited for real-time applications.

Human detector: Two-stage

Faster R-CNN (Faster Region-based Convolutional Neural Network) is an advanced algorithm developed in 2015 by Ren et al. that builds upon the R-CNN and Fast R-CNN frameworks.

- Region Proposal Network enabling the model to generate region proposals directly from feature maps.
- The RoI (Region of Interest) pooling step scales these regions to a fixed size.
- The network classifies the objects and refines their bounding box coordinates.



$\Longrightarrow$ Significant speed improvement compared to R-CNN and Fast R-CNN, high accuracy but slower speed compared to one-stage.

Pipeline overview

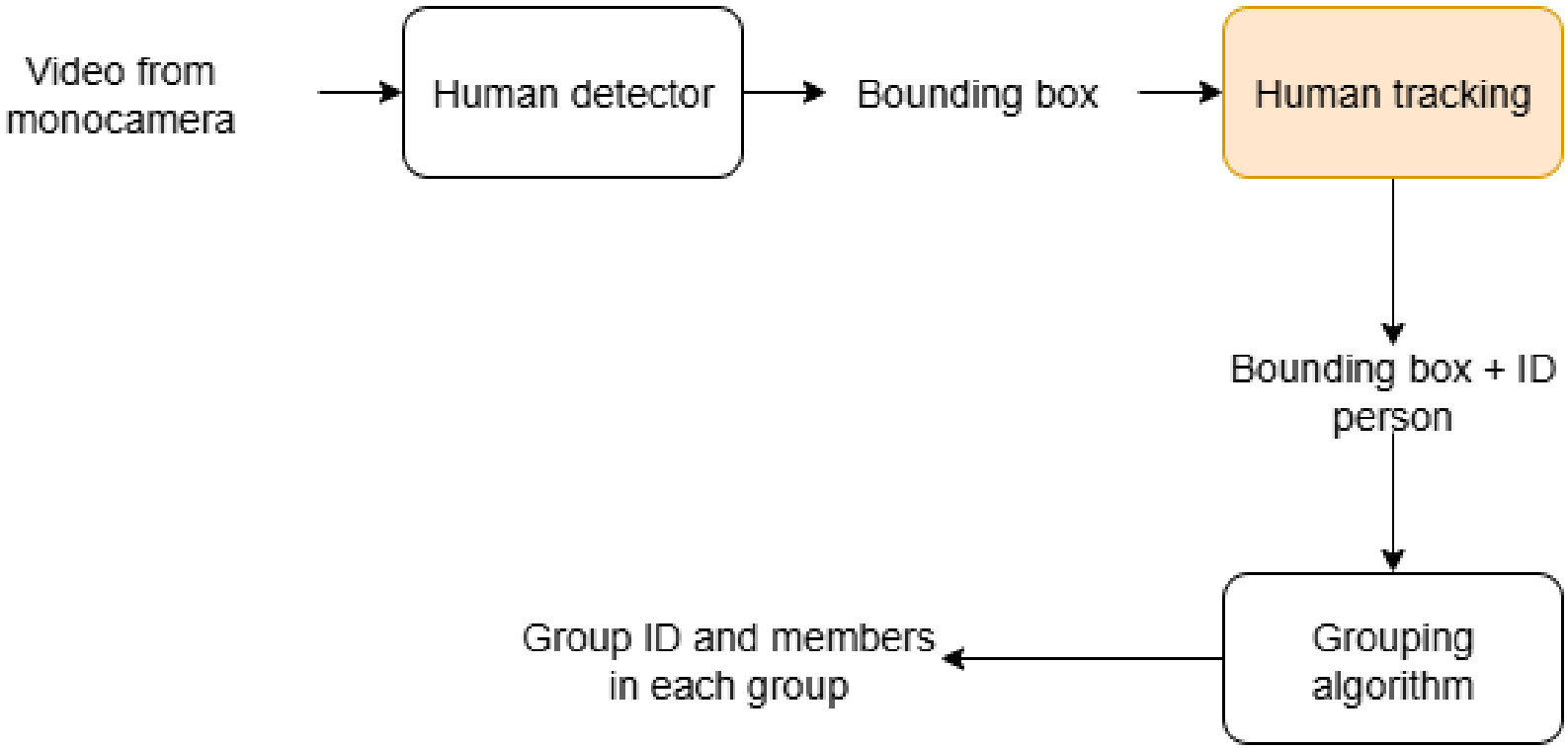


Figure: Pipeline overview of group detection and tracking system

Human tracker: Deep SORT

Deep SORT:

- Introduced by Wojke et al. in 2017.
- Kalman Filter is used to predict the next position of each track.
- Hungarian Algorithm is applied to associate new detections with the predicted tracks.
- Integrating deep appearance features (CNN) for object re-identification.

$\Longrightarrow$ Reduces identity switches and enhances tracking performance, especially in crowded or occluded scenarios.

Human tracker: Byte Track

ByteTrack:

- Introduced by Zhang et al. in 2021
- Improve tracking robustness by effectively utilizing low-confidence detections.
- High-confidence detections are matched to existing tracks using the Hungarian algorithm based on Intersection over Union (IoU) distances.
- Low-confidence detections are subsequently matched with unmatched tracks to recover potentially lost objects.

$\Longrightarrow$ ByteTrack improves by leveraging both high- and low-confidence detections, fast speed by using only IoU matching.

Human Tracker: BoT SORT

BoT SORT:

- Introduced by Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky in 2022.
- BoT-SORT is an improved version based on ByteTrack (which itself extends SORT).
- Using high- and low-confidence detections for robust tracking like Byte Track.
- BoT-SORT uses appearance features only for nearby detections to refine matching when motion/IoU is uncertain.

$\Longrightarrow$ BoT-SORT combines optimized appearance features and ByteTrack-style matching to achieve balancing tracking robustness and computational efficiency.

Pipeline overview

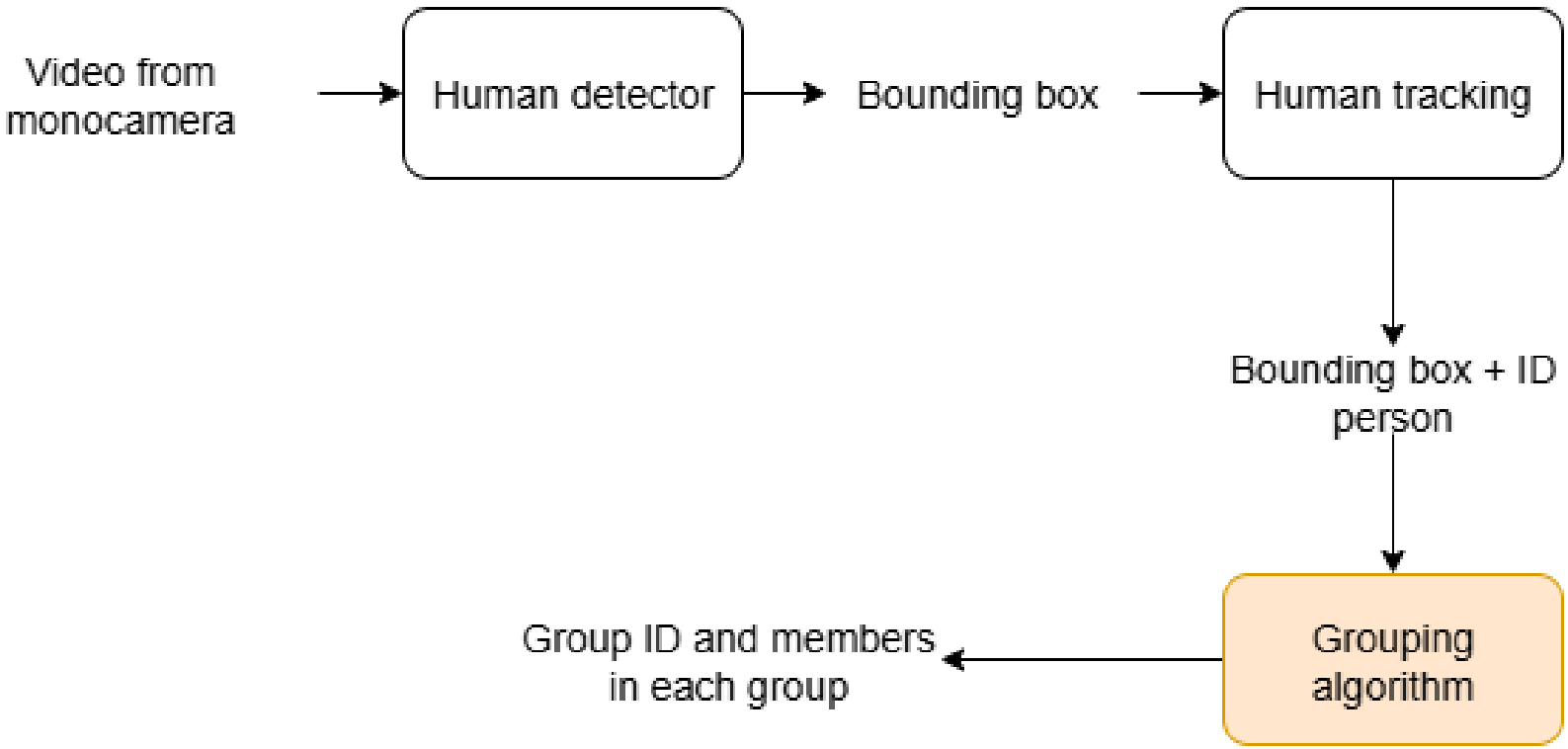
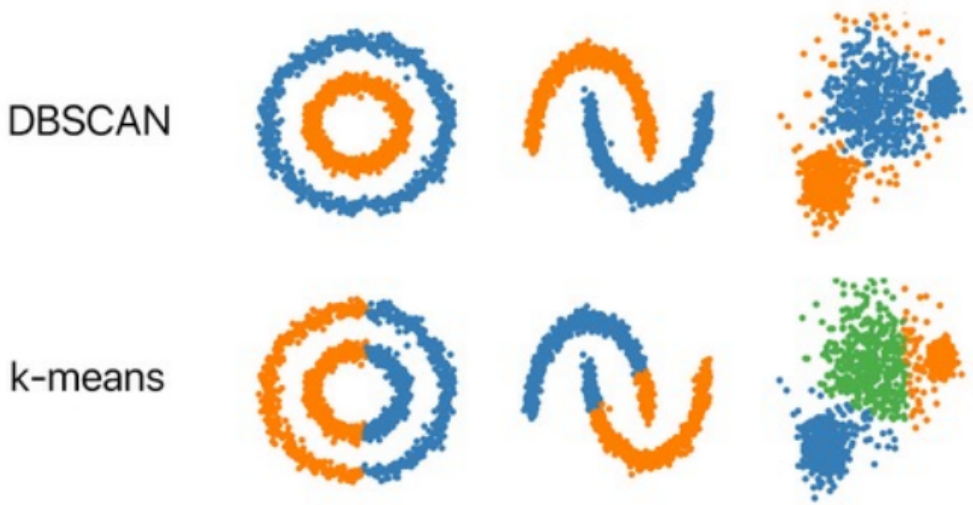


Figure: Pipeline overview of group detection and tracking system

Grouping Algorithm

- Based on DBSCAN Clustering algorithm:
  - Density-Based Spatial Clustering of Applications with Noise.
  - No need to predefine K numbers.
  - Not all points are assigned to clusters.
  - DBSCAN can find non-linear separable clusters.



Grouping Algorithm

- Based on DBSCAN Clustering algorithm.
- Incorporate an estimated depth component to extend the feature space from 2D to pseudo-3D:
  - Using the following formula[1]:

$$d = \left(\frac{2 \cdot \pi \cdot 180}{w + h \cdot 360} \cdot 1000 \right) + 3 \tag{1}$$

d: depth estimation
w, h: width, height of bounding box

  - The depth value d is inversely proportional to the sum of the bounding box width and height.

[1] Muiz Khan et al. (Oct. 2020). "An AI-Based Visual Aid With Integrated Reading Assistant for the Completely Blind". In: *IEEE Transactions on Human-Machine Systems* 50, pp. 507–517. DOI: 10.1109/THMS.2020.3027534.

Grouping Algorithm

- Based on DBSCAN Clustering algorithm.
- Incorporate an estimated depth component to extend the feature space from 2D to pseudo-3D:

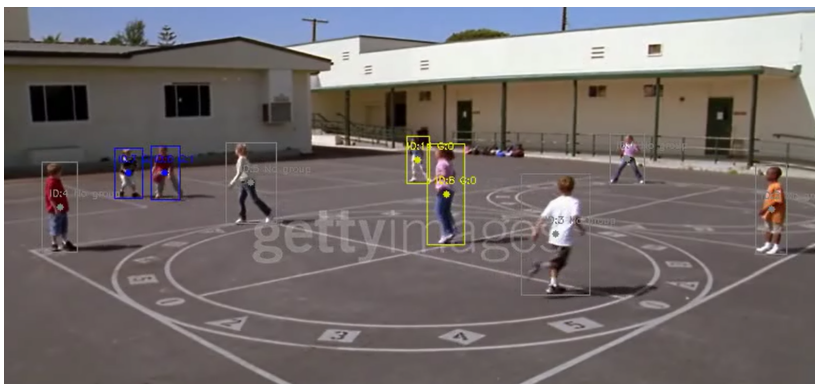


Figure: The incorrect group because of the lack of depth

Figure: The improvement after adding depth estimation

Grouping Algorithm

- Based on DBSCAN Clustering algorithm.
- Incorporate an estimated depth component to extend the feature space from 2D to pseudo-3D.
- Tracking group status across frames based on changes in members within each group:
  - The number of members retained in a group exceeds this threshold, the group is considered to be the same as in the previous frame:
    Threshold = 0.8
    Overlap = (the number of old member in new frame) / The number of member in the previous frame)
    Frame n: ID Group (ID1, ID2, ID3, ID4, ID5)
    $->$ Group ID 1
    Frame n+1: ID Group (ID1, ID2, ID3, ID4), overlap = 4/5 = 0.8
    $->$ Group ID 1
    Frame n+2: ID Group (ID1, ID6, ID7, ID8), overlap = 1/4 = 0.25
    $->$ Group ID 2

Data Video

Experimental Parameters

Experimental Pipeline:

- Pipeline 1: Faster R-CNN R50 FPN / Faster R-CNN X101 FPN + Deep SORT
- Pipeline 2: YOLOv8n + Byte Track / BoT SORT

Parameters of Grouping Algorithm:

- Epsilon: The value represents the maximum distance between two individuals for them to be considered part of the same group. [50, 75, 100]
- Min samples: The minimum number of people to form a valid group. [2]
- Overlap threshold.

Results

Figure: Some results between YOLOv8n + Byte Track (left) and Faster R-CNN R50-FPN 3X + Deep SORT (right)

Results

Figure: Some results between YOLOv8n + Byte Track (left) and Faster R-CNN R50-FPN 3X + Deep SORT (right)

Results

Figure: Some results between YOLOv8n + Byte Track (left) and Faster R-CNN R50-FPN 3X + Deep SORT (right)

Results

Table: The average inference time(s) per frame of YOLOv8 (end-to-end pipeline, include tracking + grouping)

Video	YOLOv8 + Bot Sort	YOLOv8 + ByteTrack
Video 1	0.0337	0.0207
Video 2	0.0312	0.0155
Video 3	0.0445	0.0249
Video 4	0.0357	0.0175

Table: Comparison of average inference time(s) per frame between Faster R-CNN R50-FPN and X101-FPN (Only detection, exclude tracking + grouping)

Video	Faster RCNN R50-FPN 3X	Faster RCNN X101-FPN 3X
Video 1	0.1202	0.6218
Video 2	0.1261	0.6240
Video 3	0.1253	0.6210
Video 4	0.1279	0.6237

Conclusion

Summary:

- Implemented a tracking-by-detection pipeline for subgroup identification and tracking from monovideos.
- Discover different detectors (Faster R-CNN, YOLO) with state-of-the-art trackers (DeepSORT, ByteTrack, BoT-SORT) to compare their pros and cons.
- Construct a human group detection algorithm based on DBSCAN, incorporating depth estimation to overcome the lack of depth information of monovideos.

Future works:

- Explore additional detectors and trackers to enhance system accuracy while maintaining fast inference.
- Improve subgroup detection algorithms to address limitations such as individuals passing by each other for a few frames are still mistakenly classified as a group.

Thank you for your attention!