Libraries Used:

- `scipy.spatial.distance` (for calculating Euclidean distance between centroids)
- `collections.OrderedDict` (to maintain the order of tracked objects)
- `numpy` (for efficient array operations)

⚙️ Class: `CentroidTracker`

The `CentroidTracker` class manages the lifecycle of tracked objects: registration, update, and deregistration.

Initialization

Python
```
tracker = CentroidTracker(maxDisappeared=50)
```

Parameter	Description	Default
`maxDisappeared`	The maximum number of consecutive frames an object can be missing (not detected) before it is permanently deregistered from the tracker.	50

Core Methods

Method	Description
`register(centroid)`	Assigns a new unique ID to a detected object and stores its centroid coordinates.
`deregister(objectID)`	Removes a tracked object from the tracker (used when an object disappears for too long).
`update(rects)`	The main processing function. It takes a list of bounding boxes (`rects`) from the current frame and associates them with existing tracked objects or registers them as new objects.