<< **File:** face_detection_tracking.py >>

TrackUtils

+ linear_assignment(cost_matrix) -> ndarray
+ iou_batch(bb_test, bb_gt) -> ndarray
+ associate_detections_to_trackers(detections, trackers, iou_threshold=0.3) -> Tuple[ndarray, ndarray, ndarray]
+ convert_bbox_to_z(bbox: List[float]) -> ndarray
+ convert_x_to_bbox(x: ndarray, score=None) -> ndarray

KalmanBoxTracker

+ kf: KalmanFilter
+ id: int
+ time_since_update: int
+ history: List[ndarray]
+ hits: int
+ hit_streak: int
+ age: int

+ __init__(bbox: ndarray)
+ update(bbox: ndarray)
+ predict() -> ndarray
+ get_state() -> ndarray

Sort

+ max_age: int
+ min_hits: int
+ iou_threshold: float
+ trackers: List[KalmanBoxTracker]
+ frame_count: int

+ __init__(max_age=5, min_hits=3, iou_threshold=0.3)
+ update(detections=np.empty((0, 5))) -> ndarray

CentroidTracker

+ max_disappeared: int
+ distance_threshold: float
+ next_object_id: int
+ objects: OrderedDict[int, Tuple[float, float]]
+ disappeared: OrderedDict[int, int]

+ __init__(max_disappeared=50, distance_threshold=50)
+ register(centroid: Tuple[float, float])
+ deregister(object_id: int)
+ update(centroids: List[Tuple[float, float]]) -> OrderedDict[int, Tuple[float, float]]
+ match_centroids(centroids: List[Tuple[float, float]]) -> Dict[Tuple[float, float], int]