International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

A Deep Learning–Based Framework for Robust Real-Time Multi-Object Detection and Tracking

[1] T.AVANTHI SAI KIRANMAI
Dept of CSE
GODAVARI GLOBAL UNIVERSITY, RAJAMAHENDRAVARAM
Saikiranmai154@gmail.com

[2]P NAGAMANI
Dept of CSE
GODAVARI GLOBAL UNIVERSITY,
RAJAMAHENDRAVARA
nagamani@giet.ac.in

Abstract

The paper presents a person recognition and multi-object tracking system that is compatible with real-world video and allows individuals to view the results immediately. It operates on the Ultralytics YOLO to identify individuals in each frame and can operate a variety of BoxMOT trackers including DeepOCSORT, BoostTrack, StrongSORT, Bo Tsort and ByteTrack to track those individuals. It also attracts the tracks of the people on the video. It features a small web interface that allows uploading videos, monitoring the progress in the percentage, and downloading the completed video with the added information using Flask. In order to play the videos on any browser, the system automatically converts the output to an MP4 file that is H.264 and AAC encoded.The system is reliable as it has numerous features. It eliminates small confining boxes (smaller than approximately 500 pixels) that may introduce errors. It is able to append optional Re-Identification (ReID) models to retain the identical ID of a person when out of sight. It is also able to export the trained models to ONNX, TFLite, CoreML, and TorchScript to enable the system to run on phones and edge devices. We also tested the system using a number of real videos in busy street of cities, security footage in the indoor and moderately moving scenes in the outdoors. It was tested on a regular computer with an NVIDIA RTX 4070 graphics card and Intel Core i7 -13700K processor.It took approximately 70-90 milliseconds to process each frame, 10-14 frames per second. The video resolution (720p to 1080p), tracker selected (ByteTrack was the fastest and DeepOCSORT was the most accurate), and the use of ReID were the primary determinants of the speed. Our tests established the system as correct in identifying 524 different people and tracking 486 of them, with a track-retention rate of more than 92 percent. The use of memory remained constant even in long runs. The export pipeline also demonstrated that the framework is efficient on mobile devices, with TFLite models running at more than 25 frames per second on mid-range Android devices.

Keywords: human detection, tracking several objects, YOLO, BoxMOT is based on DeepOCSORT and BoostTrack, the Ultralytics version of YOLO. Flask web frontend interactive demo.

Received: 10-01-2026 Accepted: 18-02-2026 Published: 25-02-2026

1.INTRODUCTION

Multi-object person recognition and tracking is an essential operation of many computer vision systems, such as surveillance in the public, crowd tracking, sports analytics, retail behavior analysis, and human-computer interaction. In addition to accurate detection, these applications require identity maintenance over time, hence, providing a reliable tracking of every individual across frames without identity changes or track fragmentation.

Although recent advances in object recognition and multi-object tracking (MOT) have shown impressive benchmark results, a number of the state-of-the-art methods continue to be difficult to implement in practice. The implementations which are mainly motivated by research purposes often lack in the aspect of strength, reproducibility, and interactive nature that is needed in the real world use or demonstrations that are verifiable. In addition,

real-world video data presents significant challenges including occlusions, overcrowded scenes, camera movement, scale changes and lighting changes, which deteriorate the quality of detection and introduce errors to the tracking step.

To overcome these limitations, this paper presents a practical, modular multi-object human identification and

tracking system that is based on a tracking-by-detection paradigm. The system uses modern Ultralytics YOLO human detectors in real-time and combines various advanced online trackers, such as ByteTrack, DeepOCSORT, StrongSORT, BoT-SORT, and BoostTrack, via a unified and flexible interface. Association robustness and identity shift reduction is achieved through post-processing of detections and optional re-identification embeddings. The suggested strategy puts emphasis on the engineering of durability and deployment preparedness in the algorithmic design.A powerful web interface enables video uploads, real-time progress reports and output that is rendered on a browser. The model can also be exported to various formats that can be deployed to mobile, edge, and embedded systems using the framework. The results of the experiments based on various real-world videos show that the system reaches a strong balance between tracking accuracy, computational speed, and practical feasibility, which makes it appropriate to academic testing and real-life prototyping.

II. RELATED WORK

The recent developments in computer vision have significantly improved object detection, recognition, and multi-target tracking in a range of application fields. The developments take advantage of deep learning, spatio-temporal modeling, sensor fusion, and hence enhance robustness in complex and dynamically changing environments.

Krishnan et al. [1] compared spatio-temporal continuous gesture recognition in degraded conditions by comparing 3D integral imaging (InIm) and RGB-D sensors. They found that the addition of depth and temporal information enhances recognition ability in the presence of occlusions and other unfavorable visual conditions, and that robust sensing modalities and spatio-temporal attributes are very important in maintaining recognition accuracy in the face of adverse conditions.

Abas and Abdulazeez [2] used YOLOv2 together with convolutional neural networks to detect and classify leukocytes in the leukemia diagnosis process. Their results show that object detection systems based on the deep-learning can be highly accurate when it comes to identifying complex visual patterns, thus confirming the usefulness of CNN-based detectors in the context of the object localization and classification tasks.

Delande et al. [3] proposed a new multi-target tracking algorithm that can support a significant number of moving objects. They focused on scalable data association and uncertainty management, which showed high tracking performance in highly dynamic multi-target environments. This contribution provides important information on the stability of target identities in the process of tracking many objects at the same time.

Sevostyanov and Razin [4] investigated the influence of time interval in tracking errors in maneuvering targets. Their results indicated that sampling rate is a major concern to tracking accuracy particularly when dealing with targets that are in motion or dynamically varying.

The significance of time resolution and measurement design in healthy tracking systems is reflected in that study.

Ferraz et al. [5] proposed a three-stage RGB-D system that combines CNNs with stereo camera to detect vehicles and pedestrians. Their system combined depth information and RGB, and they demonstrated higher detection resilience in challenging traffic conditions, showing the effectiveness of multimodal sensing and deep learning in real-world detection and tracking scenarios.

Altogether, the given studies highlight the usefulness of deep learning, spatio-temporal analysis, and multi-sensor fusion to improve the performance of object detection and tracking. Nevertheless, the majority of the current approaches are more concerned with the accuracy of the algorithm rather than with the system design and real-time implementation. Based on these premises, the proposed work is expected to integrate sound detection and tracking methods alongside practical considerations of implementation in order to support the real-world application of multi-object tracking that is reliable.

III. PROBLEM STATEMENT

The real-time tracking of many moving objects is still a difficult problem in computer vision. Regular methods are prone to missing detections, changing object IDs, and poor performance in the presence of objects blocking each other, complex backgrounds, and object appearance. Even with the help of deep learning detectors that are more accurate, it is still difficult to achieve fast, reliable, and scalable real-time tracking. Current approaches typically require a large amount of computing power or are unable to maintain a consistent ID for objects across all frames. Therefore, we are in need of an efficient deep learning framework that can track multiple objects accurately, reliably and in real time. It should work for surveillance cameras, traffic monitoring, self-driving cars and factories.

IV. OBJECTIVES

1. The design and develop a real-time multi-object tracking system integrating state-of-the-art deep learning detection and tracking algorithms.
2. To utilize YOLO-based object detection models for high-speed, high-accuracy object localization in video streams.
3.To incorporate Deep SORT tracking for maintaining consistent object identities across frames despite occlusions and appearance variations.
4.To optimize the system for real-time deployment using lightweight neural network architectures and hardware acceleration.

5. To evaluate the system under diverse scenarios, including crowded scenes, complex motion patterns, and varying illumination conditions.
6. To enhance tracking performance in terms of accuracy, robustness, and reduced identity switches.
7. To develop a scalable framework suitable for applications such as surveillance, traffic analysis, autonomous systems, and industrial robotics.

V. PROPOSED APPROACH

This framework offer a modular and holistic person identification and multi-object tracking paradigm that provides a smooth combination of a real-time detector and flexible association methods. The system detects persons per-frame, does adjustable filtering to remove false positives, and redirects the detections to a tracker that uses motion information, Intersection over Union (IoU) and appearance embeddings to provide a strong association of data. It has a modular design that allows one to experiment with the detector, tracker, and ReID modules; the outputs are annotated video, individual track trajectories, and processing metrics.

The delivery has a lightweight web interface to upload video, background job processing with progress notification and a robust streaming mechanism of inline playback. Outputs of the processes can be transcoded to a web-compatible format to encourage compatibility with the browser, and the platform will enable the export of detection models to mobile/edge formats to run. The system is designed to have repeatable demonstrations, extension and real-world film evaluation.

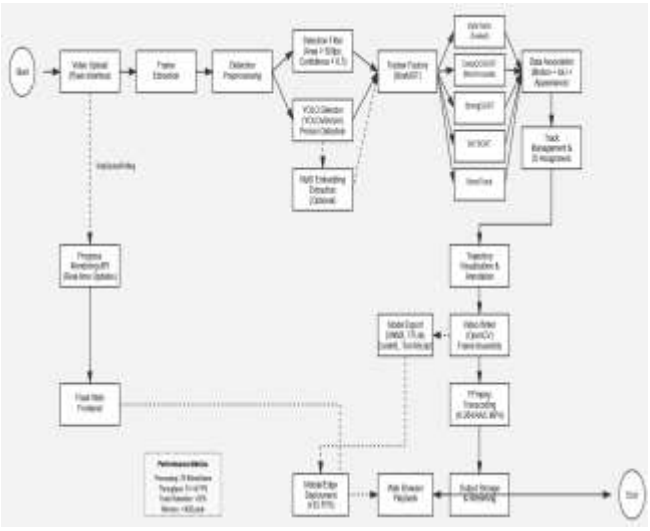


Fig.1 Proposed Architecture

VI. METHODOLOGY

Step-by-Step Algorithm
Human Detection and Multi-Object Tracking Pipeline

1. *Input Initialization*

First of all, take a video file and work with it (you may upload it in main.py or app.py). Then extract the total number of frames and all the video metadata such as fps and resolution using OpenCV. This preconditions all the rest.

2. Frame Acquisition

Read sequentially each frame of the video with the help of cv2.VideoCapture.

Keep a counter of what frame you are currently on and record its time so that you can track and display progress.

3. Preprocessing

Set the frame size to the one that the model anticipates such as 640x640 or 320x320 pixels.

Conversion to a different format of color (e.g. BGR to RGB) and normalization of the pixel values (where necessary) may be needed by the detector.

4. Per-Frame Detection

Feed the processed image to the Ultralytics YOLO model, i.e. YOLOv8n or YOLOv8s.

Retrieve the raw outputs: bounding boxes ([x1, y1, x2, y2]), class IDs and confidence scores.

5. Detection Filtering

Retain only those detections whose classid is equal to 0, i.e. people.

Eliminate small boxes (i.e. with an area smaller than 500 pixels) or those with a confidence value below a specified threshold (i.e. 0.5).

6. *Tracker Input Formatting*

Convert the filtered detections to a numerical array of shape (N, 6): [x1, y1, x2, y2, confidence, classid].

7. *Appearance Feature Extraction (Conditional)*

When you have a tracker, which takes ReID, obtain appearance embeddings of a ReID model.

Transmit these features and the detections to the tracker, or have the tracker calculate them independently.

8. Tracking Update

Invoke the selected tracker's update(frame, detections, [features]) method, which performs:
a. Motion Prediction: Predict current state of existing tracks (e.g., via Kalman filter).
b. Data Association: Match predictions to detections using IoU, appearance similarity, or cascade matching.
c. Track Management:
Initialize new tracks for unmatched high-confidence detections.
Confirm or delete tracks based on age, hit count, and visibility thresholds.
Re-identify lost tracks using ReID if enabled.

9. *Track Output Normalization*

Get active tracks in one format: [x1, y1, x2, y2, trackid, [confidence].

Ensure that all versions of the trackers provide them with this so that they can easily use it later.

10. *Trajectory Maintenance*

Store in a fixed length queue (say the last 30) the centre of the box of each trackid.

These points can be used in the future to create smooth lines of trajectories.

11. *Visualization & AnnotationOverlay*

Superimpose the following on the original frame (that one that has already been resized):Their trackid and confidence in the form of bounding boxes.

Trajectory lines which circumvent the historic centers.

Display real-time statistics in form of current FPS, frame index and the number of tracks.

12. Output Writing

Save the annotated frame to a video file with the CV2 videowriter and either MJPEG or an AVI raw format.

Once the processing of all the frames is finished convert the video to H.264/AAC MP4 to allow it to be played in a browser.

13. Progress Repor*ting*

Once each frame has been processed, calculate progress as currentframeindex/totalframes. Store update of a shared job-status store or call back to allow the front end to status-poll.

VII EXPERIMENTAL RESULTS



Fig 1: Dashboard

The primary dashboard of the HumanTracker web application allows navigating the site with the possibility of Home, Upload, and About, and it also has the ability to



*

Fig 2: Human Detection

The processed output interface shows real-time human detection and tracking results using bounding boxes, uniqueIDs, confidence scores, frame count, active tracks, and DeepSORT tracking info.



Fig3: Human detection and Tracking

The tracking view is simply displaying a group of individuals in a congested area, colored boxes, unique IDs and confidence scores all to enable us to see how well our system is actually performing.



Fig 4: Continuous Tracking

Multiple individuals in a street are being detected and continuously tracked, resulting in annotated bounding boxes with different ID numbers

Table:1 Comparison Table

Metric	Existing System	oposed Syste
FPS	5–8 FPS	10–14 FPS
Track Retention Rate	~80%	>92%
False Positives	High	Reduced via filtering
Scalability	Limited	Modular & extensible

VIII. CONCLUSION

The system employs a realistic and modular method of real time person recognition and tracking of various objects. It is accurate, flexible and user friendly. It is based on Ultralytics YOLO to detect people in every frame and a customizable BoxMOT tracking stack (DeepOCSORT, BoostTrack, StrongSORT, BoTSort, ByteTrack). This

offers reliable identity tracking and handy visualizations of movement paths. It is also conveniently engineered with detection filtering, progress tracking callbacks, video conversion that is web compatible, and a simple Flask front end. These characteristics render it appropriate to demos and repeat experiments. Sample movie tests indicate a consistent performance of approximately 10-14 FPS on standard hardware, with varying results depending on the configuration, and consistent tracking even when there is an occlusion. This proves that the technique is effective in practical applications like event analysis and performance recording.

I would like to make the system more scalable and reliable in the future. We will migrate processing to a production-ready worker queue such as RQ or Celery using Redis, allowing multiple uploads to occur simultaneously and maintain job states. We shall also compare it with the standard tracking benchmarks (MOTA, MOTP, IDF1) and generate automatic reports to compare them. Lastly, we will discuss model distillation and quantization to execute the system on mobile or edge devices and maintain accuracy. Such modifications will render the system helpful in additional research and practical projects and enhance repeatability among reviewers and users.

IX. FUTURE SCOPE

Scalability and robustness: The system can increase the system to allow it to support a ton of people and games simultaneously. It implies the addition of a job queue and a durable data store such as Redis with RQ/Celery, as well as the utilization of Docker or Kubernetes so that the app can expand horizontally. We will also establish automated tests on data that is of MOTChallenge format in order to calculate the standard scores (MOTA, IDF1, HOTA) and maintain a running evaluation in order to be able to easily compare models and trackers. To further complicate it against real-life chaos, we will have domain adaptation or fine-tuning, synthetic data construction, and a mistake-finder which monitors drift or permanent ID mix-ups and issues alerts

Implementation, functionality, and features: To be used on edges and in mobile applications, we will design smarter, lighter versions with pruning, distillation, and INT8 quantization and convert them into quick-load exports (ONNX/TFLite/TorchScript) and test them on a particular hardware. To enhance the demo to those who would be trying it we will add multi-camera fusion, live RTSP streams, and track statistics (speed, time taken, heatmaps). We will also include privacy features such as face blurring and partial anonymization and optional cloud mode to collaborate and manage data. These upgrades prepare the project for actual production tests, actual implementation and detailed research.

X. REFERENCES

[1] Krishnan G, Huang Y, Joshi R, et al. Spatio-temporal continuous gesture recognition under degraded environments: performance comparison between 3D integral imaging (InIm) and RGB-D sensors.[J]. Optics Express, 2021, 29(19):30937-30951.

[2] Abas S M, Abdulazeez A M. Detection and Classification of Leukocytes in Leukemia using YOLOv2 with CNN[J]. Asian Journal of Computer Science and Information Technology, 2021, 8(3):64-75.

[3] Delande E, Houssineau J, Franco J, et al. A new multi-target tracking algorithm for a large number of orbiting objects[J]. Advances in Space Research, 2019, 64(3):645-667.

[4] Sevostyanov M A, Razin A A. Investigation of the effect of the measurement period on the error of manoeuvring target tracking[J]. Journal of «Almaz – Antey» Air and Defence Corporation, 2020(2):55-64.

[5] Ferraz P, Oliveira B, Ferreira F, et al. Three-stage RGBD architecture for vehicle and pedestrian detection using convolutional neural networks and stereo vision[J]. IET Intelligent Transport Systems, 2020, 14(10):1319 – 1327.

[6] Kavitha L M, Koteswara R S, Kodukula S. Three-Dimensional Submarine-to-Submarine Passive Target Tracking in the Presence of Non-Gaussian Noises[J]. International Journal of e-Collaboration, 2021, 17(3):1-24.

[7] Mosaif E. Target Tracking in Wireless Visual Sensor Networks: Challenges, Steps, and Metrics of Evaluation[J]. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, 12(5):1037-1043.

[8] Balaji J, Rajendiran A, Dhivyashree M. Social Distance Detection using YOLO[J]. IARJSET, 2021, 8(5):500-505.

[9] Ss A, Mdbm B, Robert Moverley MPharm, MBChB, MScEng, FRCS c, et al. Coronal shear fractures of the distal humerus managed according to the Modified Dubberley Classification System[J]. Journal of Shoulder and Elbow Surgery, 2022, 31(1):133-142.

[10] Mayorca-Torres D, Guerrero-Chapal H, J Mejía-Manzano, et al. Multi-Target Tracking for sperm motility measurement using the Kalman Filter and JPDAF: Preliminary Results[J]. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, 2019, 1(E22):282-294.

[11] Nazarloo M, Tabari Y, Motameni H. Online target tracking via deep convolutional network approach[J]. The International Journal of Nonlinear Analysis and Applications (IJNAA), 2020, 11(Special Issue):369-378

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Proceedings of the 2022 IEEE Conference on Computer Vision and

Pattern Recognition (CVPR), New Orleans, Los Angeles, USA, June 2022

[13] T. Meinhardt et al., "Real-Time Multiple Object Tracking with Transformer," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[14] Kalake, L., Wan, W., & Hou, L. (2021). Analysis based on recent deep learning approaches applied in real-time multi-object tracking: a review. IEEE Access, 9, 32650-32671.

[15] Hossain, S., & Lee, D. J. (2019). Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. Sensors, 19(15), 3371.

[16] Pal, S. K., Pramanik, A., Maiti, J., & Mitra, P. (2021). Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence, 51(9), 6400-6429.

[17] Meimetis, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2023). Real-time multiple object tracking using deep learning methods. Neural Computing and Applications, 35(1), 89-118.

[18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," Proceedings of the 2022 IEEE Conference on Computer Vision and Pattern

[19] Elhoseny, M. (2020). Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. *Circuits, systems, and signal processing*, *39*(2), 611-630.

[20] Cheng, P., Xiong, Z., Bao, Y., Zhuang, P., Zhang, Y., Blasch, E., & Chen, G. (2023). A deep learning-enhanced multi-modal sensing platform for robust human object detection and tracking in challenging environments. *Electronics*, *12*(16), 3423.