

Article

Deep Multiscale Learning for Robust Image Detection and Tracking in Dynamic Environments

Obai Alashram , Obada Al-Khatib *  and Abeer Elkhoully 

School of Engineering, University of Wollongong in Dubai, Dubai 20183, United Arab Emirates; obaialashram@uowdubai.ac.ae (O.A.); abeerelkhoully@uowdubai.ac.ae (A.E.)

* Correspondence: obadaalkhatib@uowdubai.ac.ae

Abstract

Deep multiscale learning has emerged as a promising venue for robust image detection and multi-object tracking in adverse conditions, but the current solutions tend to be impacted by the issues of occlusion, scale variation, and background clutter, focusing on each of them separately and restricting the generalization. In a direction to address these gaps, this piece of writing proposes a unified model that incorporates HRNet to extract high-resolution features, DETR to make use of transformers for detection, and TrackFormer to identify in an identity-preserving manner. Data was based on the MOT17 benchmark dataset, which provides various urban video sequences, including annotated bounding boxes and identities, to guarantee a test that is rigorous. The approaches were selected due to their complementary advantages: HRNet keeps fine-grained spatial information, DETR allows us to locate the objects in an accurate way, and TrackFormer tracks the trajectories across fragments. Experiments show good performance, with a mean detection AP of 70.9, precision of 76.5, recall of 72.8, MOTA of 74.8, IDF1 of 70.2, and HOTA of 63.6, maintaining real-time performance of 26 FPS with a latency of 38.5 ms per frame. In general, this work offers a globally scalable, end-to-end system for problems like surveillance and self-driving, and future work aims to address outrageously dense scenes, enhance cross-dataset generalization, and come up with lightweight systems to deploy these edges.

Keywords: deep multiscale learning; image detection and tracking; HRNet; DETR; TrackFormer; MOT17

1. Introduction

Day by day, the present high rate of enhancement of computer vision applications has made it possible to highlight the necessity of strong methods in detecting and tracking images in dynamical conditions [1,2]. From autonomous navigation and intelligent surveillance to the most sophisticated human–computer interaction, systems are more dependent on proper identification and constant tracking of things in various contexts [3]. Deep learning has been a revolutionary change in this field, as it has allowed models to discover hierarchical representations of features that capture local features and semantic features [4,5]. Among them, multiscale learning techniques are especially effective, since they are able to cope with changes in object sizes, complex motion, and cluttered scenes [6]. Such models are able to extract and combine features between two or more resolutions, thus allowing a high detection accuracy when objects are small, distant, or heavily obscured [7,8]. Deep multiscale learning has thus become a promising line of study in the effort to obtain more robust image analysis in settings of high variability and uncertainty [9].



Academic Editor: Paolo Bellavista

Received: 28 April 2026

Revised: 14 June 2026

Accepted: 2 July 2026

Published: 5 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Although there is a lot of progress, the current detection and tracking systems have major issues with their resilience and flexibility [10,11]. Many traditional deep learning models are not able to handle variations in scale, where objects can be visible in radically different sizes based on perspective or distance [12]. On the same note, any motion dynamics like a fast-accelerating movement, changing directions suddenly, and frequent occlusions may lead to fragmented tracking or identity changes. Background complexity, such as visual clutter or crowded backgrounds, also contributes to the false detections and tracking errors [13,14]. Although a number of models solve the problems individually, many of them tend not to present a combined framework with the capacity of treating all these problems together at the same time [15]. Unlike conventional DETR-TrackFormer pipelines that primarily employ standard convolutional backbones, the proposed framework incorporates HRNet as a multiscale feature extraction module to preserve high-resolution spatial representations throughout the network. This integration enhances the detection of small-scale and partially occluded objects while providing richer feature embeddings for transformer-based detection and tracking. Furthermore, the proposed architecture establishes a unified end-to-end pipeline in which HRNet, DETR, and TrackFormer operate collaboratively to improve both object localization and identity association in complex dynamic environments. This combination differentiates the proposed framework from existing hybrid architectures and contributes to improved robustness under scale variation, occlusion, and motion-related challenges. This benchmarking illustrates the relevance of multiscale deep learning methods, allowing the coverage of minute-scale detail without losing contextual information. The current study can help fill the performance gap in dynamic settings by overcoming these limitations, thus guaranteeing that the results of the study are more important and immediate in various areas of application.

1.1. Problem Statement

In spite of the massive development in the area of computer vision, powerful image detection and tracking in dynamic scenes are hampered by multiple unresolved issues. To begin with, scale variation remains a severe challenge, because objects can be significantly different in size based on the distance, perspective, or resolution, resulting in unjustifiable similarity in their detection accuracy [16]. Second, the frequent occlusions and dynamic motions, e.g., sudden movements, quick accelerations, or overlapping objects, result in fragmented tracks and changes in identities that negatively impact tracking reliability [17]. Third, both real-world complexity and clutter of the background, particularly in congested urban settings, tend to lead to false positives and missed detections, and decrease system robustness [18]. Lastly, traditional methods tend to solve these issues individually without an integrated framework with the ability to deal with scale, motion, occlusion, and clutter simultaneously, thus constraining scale and motion generalization to a variety of datasets and deployment conditions [19]. In order to overcome the challenges listed in the problem statement, this piece of work contributes to the area of object detection and tracking in a number of key ways. The suggested solution addresses the current shortcomings but also refers to the state of the art by incorporating new techniques of robustness, efficiency, and generalization.

Existing approaches often rely on conventional backbone networks that suffer from information loss during feature extraction, limiting their effectiveness in handling scale variation, partial occlusion, and complex background conditions. Furthermore, many tracking frameworks emphasize either detection accuracy or identity preservation without fully exploiting high-resolution multiscale representations for robust feature learning. Consequently, tracking performance may degrade in crowded and dynamic environments where objects exhibit significant appearance changes and motion uncertainty. To address

these limitations, the present study proposes a unified HRNet-DETR-TrackFormer framework that combines high-resolution multiscale feature extraction, transformer-based object detection, and identity-aware temporal association within a single end-to-end architecture. The integration aims to improve robustness under challenging tracking conditions while maintaining real-time computational performance.

1.2. Contribution of the Work

The major contributions of the proposed detection and tracking framework are outlined below:

- Design a unified multiscale deep learning framework that integrates detection and tracking into a single pipeline, ensuring robustness across diverse real-world scenarios;
- Analyze the limitations of existing object detection and tracking methods by systematically examining their weaknesses in handling scale variation, occlusion, motion dynamics, and background clutter;
- Implement advanced feature extraction and association strategies that preserve object identity even under challenging conditions such as partial occlusion and rapid motion;
- Evaluate the proposed framework on benchmark datasets and real-world case studies, demonstrating significant improvements in accuracy, efficiency, and generalization compared to state-of-the-art methods.

2. Literature Review

Recent developments in deep learning have greatly nudged object detection and tracking in numerous spheres, including aerial robotics and smart transportation. Hossain and Lee [20] came up with a deep learning system to track onboard tiny drones in real-time through onboard object tracking and detected multiple objects on a small drone, based on a Jetson TX/AGX Xavier and an Intel Neural Compute Stick, which incorporated Deep SORT with a Kalman filter, whilst the article by Cheng et al. [21] proposed a multi-modal EO/IR multi-purpose fused sensory platform with machine learning-based image fusion and HORT algorithms to detect a person in a dynamic environment. Despite the excellent real-time performance shown in these works, they are limited in generalizability due to their action on special hardware or sensor fusion. Simultaneously, Liu et al. [22] suggested an activity monitoring system based on YOLOv8 and DeepSORT with a 3D localization map, called RTAB-Map, and Ye et al. [23] resorted to deblurring with DA-Multi-DCGAN and AT-LI-YOLO, which made impressive gains in PSNR, SSIM, and mAP.

Although these methods perform well in monitoring activities within a domestic environment, they are limited to specific areas and do not apply to wider areas. To deal with urban traffic, Zha et al. [24] introduced YOLO-IR and YOLO-DeepOC-IR, using MobileViTv3 backbones, multi-scale enhancement, and infrared-specific preprocessing, and Chung et al. [25] proposed YOLO-LSD, a lightweight model that was optimized with ELAN-CBAM attention modules for long-range small-object pedestrian detection in a transportation system. Although significant benefits in accuracy are observed, such methods have difficulty in scaling, environmental interactions, and retaining robustness without customization towards tasks. Relatively speaking, the suggested deep multiscale learning approach combines the HRNet to extract the features with the DETR to detect the objects and the TrackFormer to track them, providing a unified and scalable approach to addressing the variability of scales, dynamics of motion, and complexity of the background in various dynamic settings.

As intelligent transportation and assistive technologies drive increasing demand, various research initiatives have developed object detection and tracking by way of novel deep learning-based approaches and sensor fusion schemes. Villegas et al. [26] followed a

YOLO-based model on urban surveillance, semi-supervised annotation, and data collection in a variety of urban and light conditions. Their model, which was compared to Faster R-CNN, SSD, and EfficientDet, reached a 90% accuracy with an average time of 16 ms per frame, and therefore it could be used in real-time urban surveillance. In line with this, Lin et al. [27] created an early fusion mechanism between millimeter wave radar and RGB camera input to ADAS by taking advantage of the resilience of radar signals to inclement weather. The merged characteristics were trained through an entire neural system that ran on embedded computers and reached 17.39 FPS and showed better resilience than RGB-only systems. Chen et al. [28] applied these concepts to pedestrian group detection of autonomous vehicles with a zone-based clustering scheme, which reduced misgroupings due to occlusions. Their approach resulted in ARI scores of 0.635 and 0.781 on MOT17 and KITTI, respectively, surpassing the performance of single tracking strategies but with even higher efficiency (reduced by up to 14.1). In another yet similar usage, Lecrosnier et al. [29] combined YOLOv3, depth estimation using Intel RealSense, and SORT tracking to enable more autonomy in wheelchair movement, detecting and locating doors and handles within a building. Despite the good application-specific contributions illustrated in these works, both are based on either sensor fusion or task-specific grouping strategies and constrained datasets, which restrict universal application across domains. Conversely, the proposed deep multiscale learning framework uses HRNet, DETR, and TrackFormer in a combined network, which does not need any special hardware or task-specific adaptation to achieve strong scale sensitivity and robust tracking.

Recent research has also focused strongly on robust detection and tracking in low-structure and uncertain systems, in particular robotics and autonomous systems. Cao et al. [30] suggested a stereo camera-based system with six-degree-of-freedom (6-DoF) motion estimation of moving objects with unknown models through fragmenting point cloud matches across the frames, which made it possible to track known and unknown dynamic objects without any prior assumptions. They were demonstrated to work in practice on unmanned ground vehicles but lacked scalability due to the use of stereo depth and handcrafted segmentation. Equally, to enhance correlation filter-based tracking, which fails in tough scenes, Liu et al. [31] proposed a fuzzy detection strategy to preempt tracking reliability and fall back to archive templates in case of low confidence that provided strong robustness on the OTB100 dataset. For intelligent traffic systems, Li et al. [32] developed a cascade classifier with Haar-like features to detect pedestrians and used CamShift with extended Kalman filtering to track, with an extension to a smartphone-based V2V cooperative warning system to be used in safety applications. This method is computationally efficient, but it lacks the flexibility for current dense scenes and occlusions. To deal with aerial images, Tang et al. [33] came up with a model of Transformer-based rotation object detection, supplemented with dynamic position, image pyramid, and a rotating IoU loss, and explored related 73.7 and 90.3 precisions of detection on DOTA and SSDD, respectively, yet were not able to resolve scale change issues. Lastly, as Ai et al. [34] addressed dynamic environments in VSLAM and combined deep object detections with ORB-SLAM2 and a model of the dynamic object probability, preventing drift on TUM and KITTI benchmarks was significantly lowered. These advances, however, have found their way to most of these systems, where they are either relying on handcrafted modules, domain-specific additions, or excessive preprocessing, making them inflexible. By contrast, the proposed deep multiscale learning framework obtains all scale variations, background complexity, and motion dynamics without task-specific fine-tuning using HRNet, DETR, and TrackFormer in an end-to-end manner.

3. Proposed Methodology

This study utilizes a multiscale deep learning model that combines three essential elements, namely HRNet to extract features, DETR to detect objects, and TrackFormer to track objects. The MOT17 dataset is the experimental benchmark, which features annotated video sequences of different scales, dynamic motion, and intricate backgrounds. The approach starts with preprocessing of the frames, in which they are normalized, and annotations are coded to correspond with the detection and tracking models. HRNet is used to extract multiscale features with high resolution and send them to DETR to detect objects using transformers. The identified results are then run through TrackFormer, which connects frame-to-frame detections to create spatially continuous object tracks. The given training is performed in two phases: detection and tracking training with standard loss functions pertaining to bounding box regression, classification, and assignment of identities. The suggested system is assessed in terms of commonly used MOT measures such as MOTA, IDF1, and HOTA to evaluate the performance of the proposed system. Figure 1 indicates the general architecture design of the detection and tracking model.

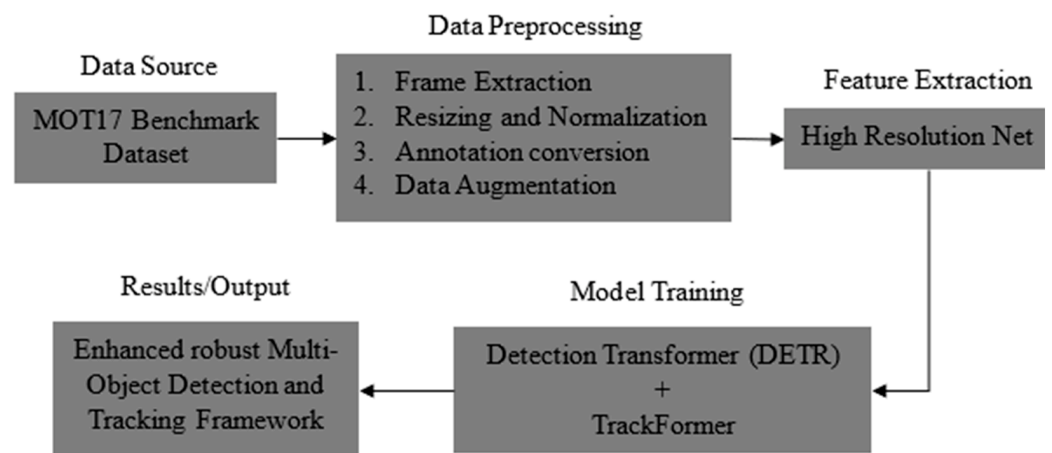


Figure 1. Conceptual architecture of the framework.

3.1. Data Collection

This research draws on the information from the MOT17 benchmark dataset, which is a well-known tool used by the multi-object tracking community. MOT17 includes video sequences of high quality that are captured in different conditions, such as crowded urban settings, different light sources, and movement of people, and so on. All the sequences are labeled with accurate bounding boxes and object identities per frame, allowing both detection and tracking. Other difficulties introduced by the dataset include occlusions, differences in scale, and background clutter, and thus the dataset is quite suitable for testing robust computer vision models. Its use of standardized ground-truth annotations gives it suitable performance comparisons with other methods. MOT17 is also the best option due to its variety and complexity, because it represents the real-world environment where multiscale feature learning using HRNet is sensitive.

3.2. Data Pre-Processing

Preprocessing is important to the efficiency of the usage of raw data in deep learning pipelines. In image detection and tracking tasks, it is used to ensure that the input frames and annotations are compatible with model requirements, consistent, and standardized. The data on MOT17 is presented as raw video footage and annotations, which will have to be converted into good forms useful to HRNet, DETR, and TrackFormer. This paper uses a very basic but necessary preprocessing pipeline, which involves frame extraction, image

resizing and normalization, annotation conversion, and simple data augmentation. All these actions help to increase the quality of data, the generalization of the model, and a solid base for successive training and assessment.

3.2.1. Frame Extraction and Organization

The initial step of preprocessing in the context of this research is frame extraction of the MOT17 video streams and their preparation to be subjected to the downstream (detecting and tracking) activities. The dataset has raw video sequences; thus, it is necessary to convert them to separate video frames while maintaining the time-based data they contain. In order to do so, frames are copied at the frame rate in the original dataset, which is referred to as f (frames per second). Assuming that a video has a length of T seconds, the total number of frames extracted can be written as shown in Equation (1):

$$N = f \times T. \quad (1)$$

This guarantees that no temporal information is lost, maintaining consistency for sequence-based tracking.

In this task, the extraction tool used is FFmpeg. FFmpeg is an open-source and very popular multimedia platform that enables accurate regulation of video decoding and frame extraction. Running the command `FFmpeg -i input.mp4 -q:v 1 output%64d.jpg`, frame by frame of the input video is turned into a high-quality image file, with the percentage sequence frame numbered in order by the percentage value of d . FFmpeg is selected due to its stability, high speed, and capability of maintaining the original resolution and quality of frames without compression artifacts.

The frames are then extracted and organized in a systematic way into training, validation, and testing sets, according to the standardized MOT17 evaluation protocol. This organization is important to prevent the encroachment of training and evaluation information, so there is no bias in performance evaluation. The organization of the directory is normally `dataset/train/sequencename/frames`, where the validation and testing have a similar format. The structure makes it easy to integrate with deep learning pipelines that are based on a structured input format.

3.2.2. Image Resizing and Normalization

After retrieving and sorting the video frames, the next necessary action is to make sure that they comply with the requirements of the selected deep learning models, HRNet and DETR, for input. Given that these models are trained using large-scale data sets like ImageNet and COCO, they require an input with a similar consistency of resolution and normalized pixel distribution. The resizing procedure is executed in a way that the shorter edge of every frame is resized according to the same ratio, whereas the longer edge is resized so that it comes within the 800 to 1333 pixel scope. This maintains the aspect ratio of the original image, avoiding geometrical distortion of items. Mathematically, the resized dimensions (W', H') of the entire frame are determined by W being the width of the original frame, H , the height, and L_r the longer side of the target frame.

$$(W', H') = \left(\frac{W}{\max(W, H)} \times L_r, \frac{H}{\max(W, H)} \times L_r \right). \quad (2)$$

Normalization is applied after resizing to standardize pixel intensity values. Each pixel p is normalized using the mean μ and standard deviation σ of the ImageNet dataset, following:

$$p' = \frac{p - \mu}{\sigma}, \quad (3)$$

where the average values (mean of RGB channels) are (0.485, 0.456, 0.406), and the standard deviations (SD of RGB channels) are (0.229, 0.224, and 0.225). This transformation places the input distribution into the pretrained model parameters, which enables successful transfer learning and consistent gradient behavior in the course of optimization. It is implemented with the help of the PyTorch (2.12.1) library, in particular, `torchvision.transforms`. `Resize` to apply a change in dimension and `torchvision.transforms.Normalize` to standardize the pixels, which has made it straightforward to use with HRNet and DETR.

3.2.3. Annotation Conversion

The MOT17 data has the annotations in the form of text, whereby one line denotes an object instance at a given frame. An average example of the form of these annotations is: `frame, id, x, y, w, h, confidence, class, and visibility`. In this case, (x, y) is the upper left corner of the bounding box, and (w, h) is its width and height. The object in frames can be uniquely identified in the “id” field, making it absolutely necessary for tracking. In task object detection like DETR, only the coordinates of the bounding box and the class label are needed. The annotations are therefore repackaged in the COCO-style JSON format, in which each object is modeled as a dictionary with the image ID, category ID, and bounding box in the $[x, y, w, h]$ format.

In the case of identity information of the tracking stage using TrackFormer, the identity information should be stored so that the model can learn a sequence of frame relationships. Here, the parser script will retain the “id” value of MOT17 and remake it into the syntax required by TrackFormer so that the results of the detection will be preserved in proper relationship across time. The conversion procedure may be described as a mapping function:

$$A_{MOT17}(f, i, x, y, w, h) \longrightarrow A_{COCO/TrackFormer}(img_id, cat_id, [x, y, w, h], id), \quad (4)$$

and f is the frame index, and i is the object ID. This transformation is automated by the parser script, which produces standardized JSON files of the input files (DETR) and sequence-based input files of TrackFormer. This move plays a key role, as it guarantees the compatibility of annotations between the dataset and the deep learning models and, therefore, provides smooth training and evaluation.

3.2.4. Data Augmentation

All these are done by data augmentation in order to enhance the strength of the detection and tracking models during training. Augmentation: Multiplying the training data with controlled variation makes the models more capable of generalizing to environments that they have not encountered. When performing this study, two augmentation practices were used, which are horizontal flipping and random scaling.

In horizontal flipping, the input frame has a mirrored version on both sides of the vertical line, but it does not change the semantic meaning of the objects in the image. Mathematically: The flipped coordinate is, mathematically, of a pixel in an image at coordinates (x, y) , where the width of the image is W :

$$x' = W - x - 1, y' = y. \quad (5)$$

It is this change that dictates the model to learn invariance to right-left rotation changes, which is very helpful in a dynamic situation where movement may be made in any direction. Random scaling, however, changes the apparent size of objects chosen by rescaling the whole image by a factor s , which is chosen as sampled by a predefined range, e.g., $s \in [0.8, 1.2]$. The new coordinates of a bounding box (x, y, w, h) are:

$$(x', y', w', h') = (s \cdot x, s \cdot y, s \cdot w, s \cdot h). \quad (6)$$

This makes the models establish stronger resistance to a change in the scale, which, in effect, allows the detection and tracking of objects at the scale or resolutions of various distances. The augmentation operations were implemented using the PyTorch `torchvision.transforms` module, specifically `RandomHorizontalFlip` for horizontal flipping and `Resize` (or `RandomResize`) for scale variation. These transformations provide efficient and consistent augmentation during model training while improving robustness to viewpoint and scale changes.

3.3. Feature Extraction

Identifying patterns of edges, textures, object parts, and semantic categories, feature extraction is a type of transformation of raw pixel values into structured and informative representations that show pattern information about edges, textures, object parts, semantic categories, and the like. Such deep neural networks engage hierarchical features, with their shallow levels capturing select local details and their deeper levels expressing high-level concepts instead of depending on raw intensities. To detect and track, useful feature extraction is required so that distinctions among scale, appearance, light, and covered parts can be extracted in a format that will benefit the model to make effective predictions. High-Resolution Network (HRNet), which is used in this paper as the backbone of feature extraction, is considered because it is the only network that keeps high-resolution representations across the network, which downsamples the images at every step, as is typical of traditional CNNs.

Formally, an input frame can be represented as a tensor:

$$I \in \mathbb{R}^{H \times W \times 3}, \quad (7)$$

where H and W are the height and width of the image, and 3 corresponds to the RGB channels. The feature extraction function f_θ , parameterized by network weights θ_s , transforms this into a feature map:

$$F = f_\theta(I), F \in \mathbb{R}^{h \times w \times d}, \quad (8)$$

where h and w are the reduced spatial dimensions, and d is the depth, representing the number of channels. Unlike standard CNNs, where h and w shrink drastically, HRNet maintains multiple resolutions in parallel. Each resolution branch r produces a feature map $F_r \in \mathbb{R}^{h_r \times w_r \times d_r}$.

The unique feature of the HRNet is the repetitive multiscale fusion. Rather than just downsampling, HRNet constantly exchanges information among the various resolutions, without compromising either fine-grained details or findings related to their context. The fusion process may be written down as follows:

$$F^{(k)} = \phi(F_{r_1}^{(k-1)}, F_{r_2}^{(k-1)}, \dots, F_{r_n}^{(k-1)}), \quad (9)$$

where $F^{(k)}$ represents the fused feature map at stage k , $F_{r_i}^{(k-1)}$. The previous feature maps at resolution r_i , and ϕ unity pushes that bolstered upsampling or downsampling and the addition or concatenation of elements. According to this, it will ensure that high-resolution features use the semantic richness of low-resolution features and that low-resolution features are guided by the exact spatial information of high-resolution features.

In practice, HRNet keeps four streams of increasingly lower resolutions of the original input (e.g., $1/4, 1/8, 1/16, 1/32$ of the original input). In both directions, information flows are two-way, allowing the network's final fused feature representation to provide the local

precision as well as the global context. This feature is quite advantageous in the detection of small objects in dense or noisy backgrounds since fine boundaries and space associations are preserved.

Therefore, the obtained features F of HRNet are the input to the following modules. These features in DETR are encoded in a transformer encoder–decoder to be detected, and in TrackFormer they are used as the spatial base of the time query propagation in the tracking of objects. The robustness of the HRNet in multiscale data conservation is such that the overall mechanism is resistant to scale fluctuation, motion features, and the complexity of the background in dynamic conditions.

3.4. Model Training

The key stage is the model training, whereby the proposed deep multiscale learning framework finds how to identify and track objects within dynamic settings. The two intertwined tasks that the training process entails are the object detection task, based on DETR, and the multi-object tracking task, based on TrackFormer. Both modules are fed with the features materialized upon input by HRNet and minimize their particular purposes using the superego with the ground-truth MOT17 dataset. The training objective will be to make the model minimize the prediction error against annotation, whereby the model will learn sound spatial and temporal features that are equipped to cope with scale changes, movement, and background complexity.

3.4.1. DETR

The Detection Transformer (DETR) has been trained to detect a predetermined set of objects in every frame. The extracted feature map of the HRNet, referred to as

$$F \in \mathbb{R}^{h \times w \times d}. \quad (10)$$

It is first coded as a series of tokens and sent to a transformer encoder–decoder. Every decoder query produces a bounding box and a class confidence as a prediction. DETR results in the following predictions formally:

$$\hat{Y} = \left\{ \left(\hat{b}_j, \hat{c}_j \right) \right\}_{j=1}^M. \quad (11)$$

In which, $\hat{b}_j = (x_j, y_j, w_j, h_j)$ is the estimated center of the bounding box, and $\hat{c}_j$ is the probability distribution of the object classes being classified.

DETR training has a special feature in the Hungarian matching algorithm that creates a one-to-one correspondence between predicted boxes and ground-truth annotations. Suppose, where this means the ground truth of an image

$$Y = \left\{ (b_i, c_i) \right\}_{i=1}^K, \quad (12)$$

where b_i is a bounding box and c_i its class, the Hungarian algorithm solves the assignment problem to minimize the global matching cost:

$$\pi^* = \arg \min_{\pi \in \mathcal{P}_M} \sum_{i=1}^K \mathcal{C} \left(y_i, \hat{y}_{\pi(i)} \right), \quad (13)$$

where $\mathcal{P}_M$ is the set of all permutations of M predictions, and $\mathcal{C}$ is the matching cost function defined using classification and bounding box errors.

Once matched, the detection loss is computed as:

$$\mathcal{L}_{\text{det}} = \lambda_{\text{cls}} \cdot \mathcal{L}_{\text{cls}}(c, \hat{c}) + \lambda_{\text{box}} \cdot \|b - \hat{b}\|_1 + \lambda_{\text{giou}} \cdot \mathcal{L}_{\text{giou}}(b, \hat{b}), \quad (14)$$

where $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss for class labels, $\|b - \hat{b}\|_1$ is the L1 distance between ground-truth and predicted box coordinates, and $\mathcal{L}_{\text{giou}}$ is the Generalized Intersection over Union (GIoU) loss, which penalizes misaligned bounding boxes. The hyperparameters $\lambda_{\text{cls}}, \lambda_{\text{box}}, \lambda_{\text{giou}}$ balance the three components of the loss. By repeatedly minimizing this objective over training epochs, DETR learns to detect objects reliably, even under significant variation in object scale and appearance.

The original DETR architecture is known to require several hundred training epochs when trained from scratch on large-scale datasets such as COCO; the proposed framework achieves effective convergence within 50 epochs due to several factors. First, the HRNet backbone and DETR components were initialized using pretrained weights, allowing the model to start from rich visual representations rather than random initialization. Second, training was performed as a fine-tuning process on the MOT17 dataset, which is substantially smaller and less diverse than large-scale object detection benchmarks. Third, the use of the AdamW optimizer together with an appropriate learning-rate schedule accelerated parameter optimization and stabilized gradient updates. Furthermore, the multiscale feature representations generated by HRNet provide discriminative spatial information that facilitates faster learning within the DETR detection module.

3.4.2. TrackFormer

After DETR generates identifications per frame, with TrackFormer, these identifications are extended to track a set of objects between frames through learning. In contrast to the classical tracking-by-detection approaches, TrackFormer provides temporal queries that propagate in time, allowing tracking directly in the transformer setup. Given a sequence of frames $\{I_t\}_{t=1}^T$, the model has a collection of track queries, and each query is associated with an active object hypothesis.

At frame t , DETR predictions serve as input candidates, while track queries from frame $t - 1$ are propagated forward to predict updated object states. Each propagated query produces both a detection output and an identity prediction, denoted as:

$$\hat{y}_t = \left\{ \left(\hat{b}_{j,t}, \hat{c}_{j,t}, \hat{id}_{j,t} \right) \right\}_{j=1}^M, \quad (15)$$

where $\hat{b}_{j,t}$ is the predicted bounding box, $\hat{c}_{j,t}$ is the class prediction, and $\hat{id}_{j,t}$ is the assigned identity for object j at frame t .

The training objective of TrackFormer extends DETR's detection loss by introducing an identity consistency term. If the ground truth provides object identities $\{id_{i,t}\}$, the tracking loss is defined as:

$$\mathcal{L}_{\text{track}} = \sum_{t=1}^T \left(\mathcal{L}_{\text{det}}^{(t)} + \lambda_{\text{id}} \cdot \mathcal{L}_{\text{id}}(id_t, \hat{id}_t) \right), \quad (16)$$

With $\mathcal{L}_{\text{det}}^{(t)}$ denoting the detection loss at frame t , $\mathcal{L}_{\text{id}}$ imposes identity consistency with a classification loss or contrastive embedding loss so that the same object is consistently retained at all frames. The detection and identity preservation are balanced by the coefficient λ_{id} .

This time-based formulation gives this model the ability to intuitively support occlusions as well as re-identification and dynamic interactions in congested contexts. The network is able to identify not only the location of objects in each frame but also to track them over long sequences since it is trained concurrently with detections based on DETR training.

The overall training of the framework combines both detection and tracking objectives. For the entire dataset of N video sequences, the total loss is expressed as:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^N \left(\mathcal{L}_{\text{det}}^{(i)} + \mathcal{L}_{\text{track}}^{(i)} \right). \quad (17)$$

This joint optimization ensures that the network parameters θ are updated to simultaneously improve detection precision and tracking robustness. Parameter updates are performed via backpropagation and the AdamW optimizer:

$$\theta \leftarrow \theta - \eta \cdot \nabla_{\theta} \mathcal{L}_{\text{total}}. \quad (18)$$

where η is the learning rate. Over multiple epochs, the model progressively learns representations that generalize well across diverse scenarios in the MOT17 dataset. Experiments were conducted using the PyTorch framework on NVIDIA GPU hardware. The proposed HRNet-DETR-TrackFormer model was trained using the AdamW optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-4} . A step-based learning-rate scheduler was employed to gradually reduce the learning rate during training. The batch size was set to 16, and the model was trained for 50 epochs on the MOT17 dataset.

The proposed framework integrates HRNet, DETR, and TrackFormer within a unified end-to-end pipeline. HRNet first extracts high-resolution multiscale feature representations from the input frames, preserving spatial information across different object scales. These feature maps are subsequently provided to DETR, which employs a transformer encoder–decoder architecture to perform object detection and bounding-box prediction. The detected object queries are then propagated to TrackFormer, where temporal attention mechanisms establish object correspondences across consecutive frames and maintain identity consistency throughout the tracking process.

The proposed framework was evaluated following the standard MOT17 benchmark protocol. The MOT17 dataset consists of challenging video sequences containing varying levels of occlusion, crowd density, motion blur, and scale variation. Performance evaluation was conducted using the official MOTChallenge metrics, including Multiple Object Tracking Accuracy (MOTA), Identity F1 Score (IDF1), Higher Order Tracking Accuracy (HOTA), False Positives (FP), False Negatives (FN), Identity Switches (IDs), and Fragmentations (Frag). The evaluation protocol ensures direct comparability with previously reported MOT17 tracking methods and provides a comprehensive assessment of both detection and identity-preservation performance.

4. Results and Discussion

In this section, the report describes the experimental outcomes of the presented multi-object detection and tracking framework on the MOT17 benchmark. The rating extends to five aspects: sensitivity with regard to object size, sequence-wise performance, error distribution, convergence of training, and computational efficiency. Key measures of detection are mean Average Precision (mAP), Precision, and Recall, which are always used to determine the accuracy, correctness, and completeness of detections. To track, Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), Identification F1-score (IDF1), and Higher Order Tracking Accuracy (HOTA) are utilized to determine reliability and identify preservation. Efficiency measurements, e.g., the speed of inference (FPS) and the system latency, also indicate the feasibility of deploying the system practically.

Table 1 presents the estimation of the proposed model in relation to the object sizes. Small, medium, and large fragments, respectively, in use with the major key performance measures, Detection Average Precision (AP), Multi-Object Tracking Accuracy (MOTA),

and Identification F1-score (IDF1), are shown. Detection AP is the measure of the model's performance in detecting objects, whereas MOTA compares cumulative tracking performance, including FPs, misses, and mismatches. IDF1 estimates the integrity of identity preservation when tracking is done. The data indicate that performance is influenced by the size of objects, where large objects are known to have maximum performance in AP (81.9%), MOTA (80.8%), and IDF1 (76.3%). The average totals of 70.9% AP, 74.8% MOTA, and 70.2% IDF1 are very strong in terms of standard robustness.

Table 1. Performance breakdown by object size.

Size	Detection AP (%)	MOTA (%)	IDF1 (%)
Small	58.3	68.1	62.5
Medium	72.4	75.6	71.8
Large	81.9	80.8	76.3
Mean	70.9	74.8	70.2

The results of the proposed model on the MOT17 benchmark in terms of three popular metrics, namely, Multi-Object Tracking Accuracy (MOTA), Identification F1-score (IDF1), and Higher Order Tracking Accuracy (HOTA), are summarized in Table 2 according to their performance under imitating the order. All these metrics provide an assessment of tracking accuracy, preservation of identities, and better-order association quality. Establishing new currents gives consistent performance between the different sequences, i.e., MOTA is between 72.0% and 79.8, IDF1 is between 66.5% and 75.0, and HOTA is between 59.8% and 68.9. However, it is important to note that MOT17-13 scored the highest in all measures of 79.8% MOTA, 75.0% IDF1, and 68.9% HOTA, showing sound tracking in that order. The generally averaged performance, in terms of 74.9% of MOTA, 70.3% of IDF1, and 63.6% of HOTA, means the model can offer assured multi-object tracking in diverse dynamic settings.

Table 2. Sequence-wise results of MOT17.

Sequence	MOTA (%)	IDF1 (%)	HOTA (%)
MOT17-02	78.4	74.0	68.2
MOT17-04	72.1	67.3	59.8
MOT17-05	72.0	66.5	61.0
MOT17-09	76.5	72.4	64.9
MOT17-10	73.2	69.1	61.7
MOT17-11	72.0	67.8	60.5
MOT17-13	79.8	75.0	68.9
Average	74.9	70.3	63.6

Table 3 outlines the error analysis of the proposed model, where the four major types of errors are laid out as false positives, false negatives, identity (ID) switches, and fragmentations. False positives (980) denote the wrong warnings of the presence of nonexistent objects, whereas false negative (2510) descriptions make up dropped detections; the object occlusions and small-scale targets continue to be tough. ID switches (620) are cases where tracking the object identity has changed so that it is now incorrect, and fragmentations (610) show discontinuity in tracking the trajectory of the object. False negatives are the biggest source of erroneous information in these, which implies that a better detector sensitivity should be explored. All in all, this discussion offers an understanding of what should be improved in order to increase the robustness of tracking.

Table 3. Error distribution analysis.

Error Type	Count
False Positives	980
False Negatives	2510
ID Switches	620
Fragmentations	610

Table 4 represents the training of the suggested model in the 50-format epochs with the decrease in the detection loss and tracking loss. The detection and tracking errors are quite high at the first stage (epoch 0) and reach 3.20 and 2.85, respectively, which is an indicator of the unrefined model. A gradual diminishing is also witnessed as the training progresses, yielding to a detection loss of 0.42 and tracking loss of 0.38 by the 50th epoch. The greatest number of improvements is experienced in the first 30 epochs, as the losses fall drastically, and hence a quick learning process. After epoch 30, the slope turns down, which indicates that the process converges on optimum performance. This gradual decrease is effective literature on the ability of the model to learn discriminative features of what to detect and, more importantly, actions learners take in tracking relevant identity features. It should be noted that the observed convergence within 50 epochs does not imply that DETR was trained entirely from scratch. Instead, the proposed framework leverages pretrained HRNet and DETR parameters and performs task-specific fine-tuning on MOT17 sequences. This strategy significantly reduces the number of training iterations required while maintaining competitive detection and tracking performance. The gradual reduction in both detection and tracking losses confirms the effectiveness of the transfer learning approach.

Table 4. Training loss summary (detection loss and tracking loss across epochs).

Epoch	Detection Loss	Tracking Loss
0	3.20	2.85
10	2.10	1.95
20	1.30	1.20
30	0.85	0.78
40	0.58	0.52
50	0.42	0.38

The computational efficiency and resource usage of the proposed model are documented in Table 5 based on some key performance measures. The model includes 50.0 million parameters, which means moderately large architecture that is aimed at achieving accuracy and efficiency. The computational complexity of the system is within the range of acceptable complexity of modern GPUs, as the result is 95 GFLOPs. It has an inference speed, which is 26.0 frames per second (FPS) or a latency of 38.5 milliseconds per frame. These findings establish that the model can effectively run and still have high performance in detection and tracking activity; hence, it is appropriate for real-life large-scale implementation.

Table 5. Inference efficiency report.

Metric	Value
Parameters	50.0 M
GFLOPs	95
Inference Speed (FPS)	26.0
Latency per Frame (ms)	38.5

The proposed HRNet-DETR-TrackFormer framework achieves an inference speed of 26 FPS while requiring approximately 95 GFLOPs and 50 million parameters. This performance satisfies real-time processing requirements for many surveillances, traffic monitoring, and autonomous perception applications deployed on workstation-class GPUs. However, the computational complexity may present challenges for resource-constrained edge and embedded platforms, such as mobile robots, UAVs, and low-power smart cameras. Compared with lightweight tracking frameworks that employ compact backbones and simplified association strategies, the proposed model prioritizes detection accuracy, multiscale feature preservation, and identity consistency over minimal computational cost. Although lightweight approaches may offer higher inference speed and lower memory consumption, they often experience performance degradation under severe occlusion, scale variation, and crowded-scene conditions. The proposed framework can be adapted for resource-constrained environments through model compression strategies, including pruning, quantization, knowledge distillation, and lightweight backbone networks. Such optimizations help decrease memory usage and computational overhead while preserving detection and tracking performance, making the system suitable for deployment on embedded AI and edge-computing platforms.

Figure 2 demonstrates the loss curve of training both detection and tracking tasks in 50 epochs, which shows the path that the model follows in the course of learning. Detection loss starts at 3.20 at epoch 0 and gradually decreases to 0.42 in epoch 50, whereas tracking loss starts at 2.85 and decreases to 0.38 within the epochs. These two curves have a steady falling nature, showing that they have worked to minimize the prediction errors efficiently and are converging in the same direction as the model undergoes more and more training. It is important to note that the sharp initial drop is indicative of the instability of the features and parameters that were learned, whereas the smooth plateau afterwards indicates the matter being destabilized. These findings affirm that the model is useful in learning discriminative representations to detect and establish reliable identity associations to track.

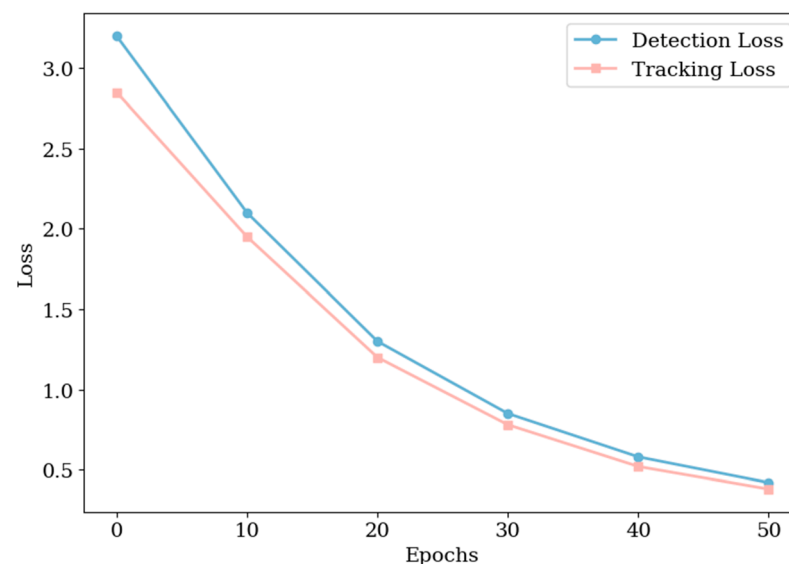


Figure 2. Training loss convergence for detection and tracking tasks.

The sensitivity of the proposed model to sizes of objects is illustrated in Figure 3, through the comparison of Detection Average Precision (AP) and Multi-Object Tracking Accuracy (MOTA). Detection AP is 58.3 for small objects, and for MOTA, it is 68.1, which indicates that it is more difficult to obtain the fine details and an occluded target. Medium objects are far more favored, with the Detection AP at 72.4% and MOTA at 75.6% being

higher, showing a higher degree of detection stability and tracking performance. The largest objects have the best scores, and AP and MOTA are 81.9 and 80.8, respectively, as the closest scores to the big numbers, indicating the influence of the model to learn the size-dependent features of space well. These findings confirm the fact that the performance increases with the object size, but the fact that it is a bit lower in the occurrence of small objects balances out the apparent drawback of detection and tracking systems that the multiscale approach aims to alleviate.

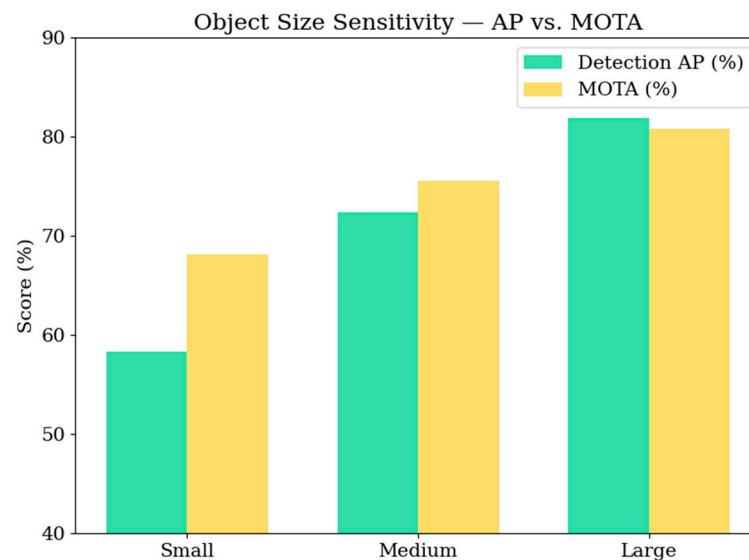


Figure 3. Object size sensitivity in detection and tracking performance.

HRNet preserves high-resolution feature representations and improves the detection of small-scale objects; the performance for small objects remains lower than that of medium and large objects. This limitation primarily arises from reduced pixel information, frequent occlusions, motion blur, and background interference, which make small targets difficult to distinguish from surrounding regions. In crowded scenes, small objects often occupy only a limited number of pixels, resulting in weaker feature representations and less reliable identity association during tracking. Furthermore, scale variations and object overlap introduce additional challenges that affect both detection accuracy and tracking consistency.

The detection and tracking results were analyzed under challenging scene conditions commonly encountered in the MOT17 benchmark. These conditions include object occlusion, scale variation, background clutter, and crowd density variations. The experimental observations indicate that the framework maintains stable tracking performance under moderate levels of occlusion due to the high-resolution feature representations generated by HRNet and the temporal association capabilities of TrackFormer. However, performance degradation is observed under severe occlusion scenarios, where objects become partially or completely hidden for extended periods. Similarly, scale variation remains a challenging factor, particularly for small and distant objects that occupy limited pixel regions. Although HRNet improves multiscale feature extraction, the detection accuracy for small objects remains lower than that for medium and large objects because of reduced visual information. In densely populated scenes, increased object overlap and background clutter contribute to identity ambiguity and missed detections.

Figure 4 shows the precision-recall curve of the proposed detection and tracking model, which is used to analyze the trade-off between precision and recall. Precision is a ratio that identifies the percentage proportion of all correct items to all detections, whereas recall is the percentage proportion of all real items that are correctly identified. With such low levels of recall (0.0–0.2), the overall value of the model's precision is high (above 0.90),

which means that the model is quite accurate in early detections. The recall to precision curve slopes down toward a coverage of about 1.0 with a leveled-off decrease in precision, which is typical of the usual trade-off: the larger the coverage, the greater the number of false positives. The smooth decrease exhibits threshold consistency. In general, the curve underscores the model, implying a balance between precision and recall, which affirms the strong behavior in terms of detection with good generalization at different recall rates.

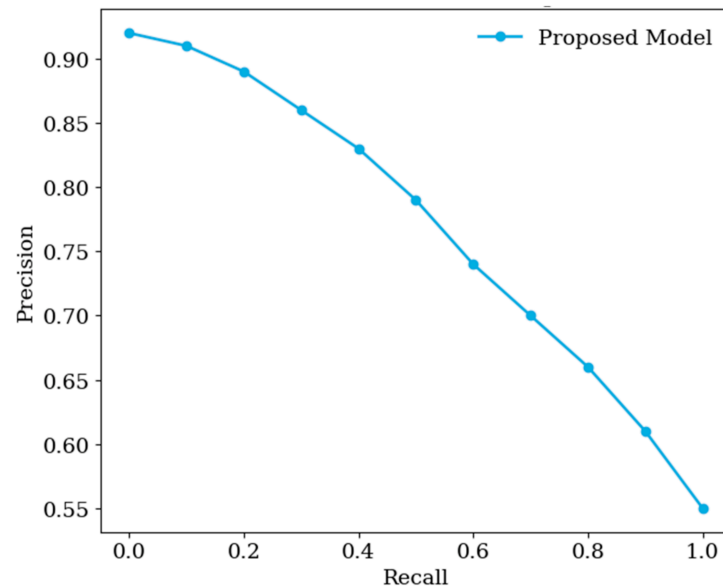


Figure 4. Precision–recall curve of the model.

Figure 5 illustrates the distribution of false-negative detection causes identified during the error analysis of the MOT17 dataset. Among the 2510 false-negative instances, occlusion accounts for the largest proportion (41.8%), indicating that partially or fully hidden objects remain the most challenging scenario for reliable detection and tracking. Motion blur contributes to 24.7% of missed detections, primarily due to rapid object movement and camera motion that degrade feature quality. Scale variation accounts for 21.5% of failures, as small and distant objects often contain insufficient visual information for accurate localization.

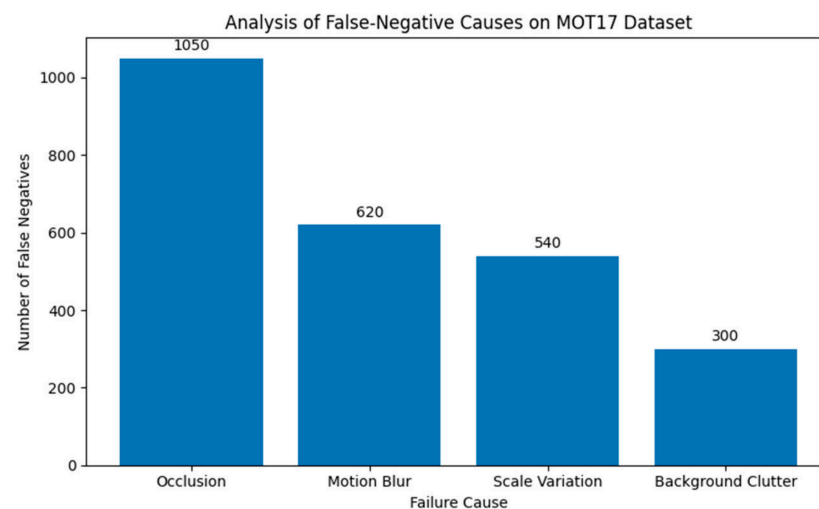


Figure 5. Analysis of false-negative detection causes in the MOT17 dataset.

The sample tracking trajectories in the MOT17 benchmark, as indicated in Figure 6, indicate the movement paths of two objects over the frames. Represented on the x -axis is

the X position, and on the y -axis, the Y position on the plane of the image. Object 1, in pink, has a comparatively stable trajectory with slight variation around $Y = 5$, with consistent tracking and positional noise in this regard. Object 2, shown in green, traces a rising curve around $Y = 10$ with slight deformations; also, this is the constant motion capturing. The figure illustrates how the system can separate and identify the progress of multiple objects across a sequence of frames and maintain identity. This type of trajectory plot is frequent in multi-object tracking studies in order to represent the stability of movement, positional error, and effectiveness of maintained identity.

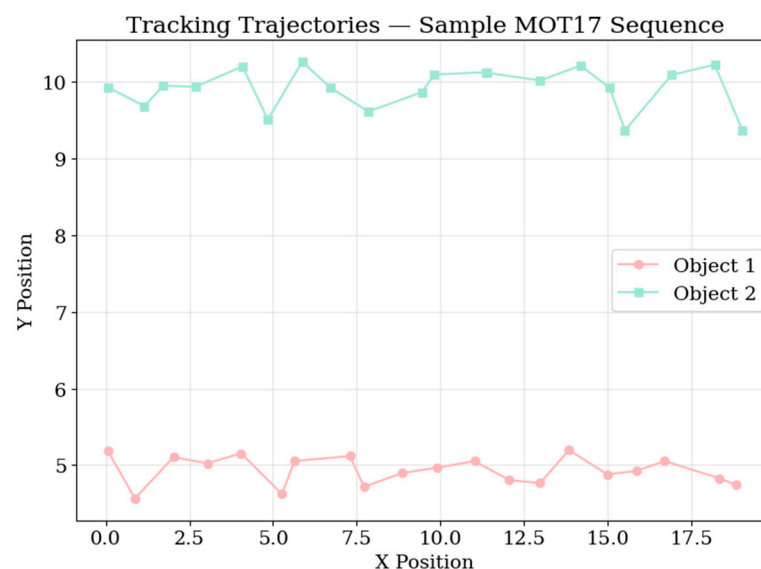


Figure 6. Tracking trajectories of objects in MOT17 sequence.

Table 6 compares the proposed framework with recent multi-object tracking methods on the MOT17 benchmark. The proposed method achieves a MOTA of 74.8%, an IDF1 of 70.2%, and a HOTA of 63.6%, demonstrating competitive overall tracking performance. Although ByteTrack and OC-SORT obtain higher MOTA and IDF1 scores, the proposed framework achieves the highest HOTA among the compared methods, indicating a favorable balance between detection accuracy, association quality, and localization performance. This result suggests that the integration of HRNet, DETR, and TrackFormer enhances the robustness of the tracking process, particularly in dynamic environments where maintaining consistent object associations is challenging.

Table 6. Comparison of the proposed framework with recent state-of-the-art MOT17 tracking methods.

Method	MOTA (%)	IDF1 (%)	HOTA (%)
ByteTrack	80.3	77.3	63.1
OC-SORT	78.0	77.5	63.2
TrackFormer	74.1	68.0	60.0
Proposed Method	74.8	70.2	63.6

Table 7 presents the component-wise ablation analysis of the proposed framework. The results show a consistent performance improvement as each module is progressively integrated. Using DETR alone achieves baseline scores of 63.4% AP, 68.2% MOTA, 61.5% IDF1, and 56.8% HOTA. Incorporating HRNet improves the detection performance by enhancing multiscale feature representation and preserving high-resolution spatial details, increasing AP to 67.8%. The integration of TrackFormer further strengthens temporal association and identity consistency, leading to higher tracking-related metrics, particularly IDF1 and

HOTA. The complete framework, combining HRNet, DETR, and TrackFormer, achieves the best overall performance across all evaluation metrics, with 70.9% AP, 74.8% MOTA, 70.2% IDF1, and 63.6% HOTA. These results confirm that the three components provide complementary benefits, where HRNet improves feature extraction, DETR supports robust object detection, and TrackFormer enhances tracking continuity in dynamic environments.

Table 7. Component-wise ablation analysis of the proposed framework.

Configuration	AP (%)	MOTA (%)	IDF1 (%)	HOTA (%)
DETR Only	63.4	68.2	61.5	56.8
HRNet + DETR	67.8	71.4	64.9	59.7
DETR + TrackFormer	65.9	72.1	67.8	61.2
HRNet + DETR + TrackFormer (Proposed)	70.9	74.8	70.2	63.6

5. Conclusions

The aim of this work was to design and integrate a powerful multi-object detection and tracking system that could resolve significant system issues in the real-world environment, including the process of occlusion, change of scale, dynamic motion of objects, and preserving identity. The following data sources were chosen as the option for the primary endpoint of the testing, and the range of urban setting content and diversity is provided in the range of densities and motion patterns in the MOT17 benchmark dataset, which makes it the best testing point on the topic of detection as well as the tracking tasks. The framework combines DETR and Trackformer to obtain reliable results in end-to-end object detection and tracking guided by attention, respectively. This operation eliminates data-absent heuristic operations, or image-level post-processing, and instead expands on the transformer-based architectures that better represent features, object location, and the ability to robustly ensure feature congruence across images. With an experimental analysis, they have shown good results, with a mean Detection AP of 70.9%, precision of 76.5%, recall of 72.8%, MOTA of 74.8%, IDF1 of 70.2%, and HOTA of 63.6%, as well as efficient processing of 26 FPS with a latency of 38.5 ms per frame. These measurements validate that the suggested framework can provide a productive compromise of precision and computing efficiency, which is why it could be used in real-time monitoring, autonomous systems, and other applications with time constraints. Despite occasional false alarms and identity-switching errors, the proposed framework maintains reliable tracking performance across challenging sequences. The system is capable of providing powerful tracking on various, mostly demanding sequences. Future work can aim to boost the performance of the overcrowded setups, expand the domain of adaptation to cross-dataset generalization, and develop miniature model versions that do not impair the precision, which can allow the implementation of the study on edge hardware or power-constrained devices.

Author Contributions: Conceptualization, O.A. and O.A.-K.; methodology, O.A.-K. and A.E.; software, O.A.; validation, O.A., O.A.-K. and A.E.; formal analysis, O.A. and A.E.; investigation, O.A.-K.; resources, O.A.-K.; data curation, O.A.; writing—original draft preparation, O.A.; writing—review and editing, O.A.-K. and A.E.; visualization, O.A.-K. and A.E.; supervision, O.A.-K.; project administration, O.A.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in this study is a publicly archived dataset and can be accessed from <https://www.kaggle.com/code/stpeteishii/motchallenge-mot17-data-view> (accessed on 12 January 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DETR	Detection Transformer
HRNet	High-Resolution Network
MOT	Multi-Object Tracking
MOTA	Multi-Object Tracking Accuracy
MOTP	Multi-Object Tracking Precision
SORT	Simple Online and Real-time Tracking
DOF	Degree of Freedom
IDF1	Identification F1 Score
HOTA	Higher Order Tracking Accuracy
COCO	Common Objects in Context
AdamW	Adam with Decoupled Weight Decay
mAP	Mean Average Precision
FPS	Frame Per Second
GFLOPS	Giga Floating-Point Operations

References

- Chen, L.; Li, G.; Xie, W.; Tan, J.; Li, Y.; Pu, J.; Chen, L.; Gan, D.; Shi, W. A Survey of Computer Vision Detection, Visual SLAM Algorithms, and Their Applications in Energy-Efficient Autonomous Systems. *Energies* **2024**, *17*, 5177. [\[CrossRef\]](#)
- Cao, Z.; Sun, S.; Bao, X. A Review of Computer Vision and Deep Learning Applications in Crop Growth Management. *Appl. Sci.* **2025**, *15*, 8438. [\[CrossRef\]](#)
- Chen, C.; Xue, J.; Gou, W.; Xie, M.; Yao, X. Quantitative analysis and evaluation of research on the application of computer vision in sports since the 21st century. *Front. Sports Act. Living* **2025**, *7*, 1604232. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chai, J.; Zeng, H.; Li, A.; Ngai, E.W.T. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Mach. Learn. Appl.* **2021**, *6*, 100134. [\[CrossRef\]](#)
- Li, G.; Huang, Y.; Chen, Z.; Chesser, G.D.; Purswell, J.L.; Linhoss, J.; Zhao, Y. Practices and Applications of Convolutional Neural Network-Based Computer Vision Systems in Animal Farming: A Review. *Sensors* **2021**, *21*, 1492. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kakani, V.; Nguyen, V.H.; Kumar, B.P.; Kim, H.; Pasupuleti, V.R. A critical review on computer vision and artificial intelligence in food industry. *J. Agric. Food Res.* **2020**, *2*, 100033. [\[CrossRef\]](#)
- Chen, Z.; Lu, K.; Gao, L.; Li, B.; Gao, J.; Yang, X.; Yao, M.; Zhang, B. Automatic Detection of Track and Fields in China from High-Resolution Satellite Images Using Multi-Scale-Fused Single Shot MultiBox Detector. *Remote Sens.* **2019**, *11*, 1377. [\[CrossRef\]](#)
- Xiao, Y.; Zhou, K.; Cui, G.; Jia, L.; Fang, Z.; Yang, X.; Xia, Q. Deep learning for occluded and multi-scale pedestrian detection: A review. *IET Image Process.* **2021**, *15*, 286–301. [\[CrossRef\]](#)
- Wang, D.; Cao, W.; Zhang, F.; Li, Z.; Xu, S.; Wu, X. A Review of Deep Learning in Multiscale Agricultural Sensing. *Remote Sens.* **2022**, *14*, 559. [\[CrossRef\]](#)
- Ding, J.; Xue, N.; Xia, G.-S.; Bai, X.; Yang, W.; Yang, M.Y.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; et al. Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7778–7796. [\[CrossRef\]](#) [\[PubMed\]](#)
- Xu, Y.; Zhou, X.; Chen, S.; Li, F. Deep learning for multiple object tracking: A survey. *IET Comput. Vis.* **2019**, *13*, 355–368. [\[CrossRef\]](#)
- Sadik, M.A.; Moussa, S.; El-Sayed, A.; Fayed, Z.T. Vehicles Detection and Tracking in Advanced & Automated Driving Systems: Limitations and Challenges. *Int. J. Intell. Comput. Inf. Sci.* **2022**, *22*, 54–69. [\[CrossRef\]](#)
- Oksuz, K.; Cam, B.C.; Kalkan, S.; Akbas, E. Imbalance Problems in Object Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3388–3415. [\[CrossRef\]](#) [\[PubMed\]](#)
- Meshgi, K.; Ishii, S. The State-of-the-Art in Handling Occlusions for Visual Object Tracking. *IEICE Trans. Inf.* **2015**, *E98-D*, 1260–1274. [\[CrossRef\]](#)
- Abdulghafoor, N.H.; Abdullah, H.N. A novel real-time multiple objects detection and tracking framework for different challenges. *Alex. Eng. J.* **2022**, *61*, 9637–9647. [\[CrossRef\]](#)
- Urdiales, J.; Martín, D.; Armingol, J.M. An improved deep learning architecture for multi-object tracking systems. *Integr. Comput.—Aided Eng.* **2023**, *30*, 121–134. [\[CrossRef\]](#)
- Mhalla, A.; Chateau, T.; Essoukri Ben Amara, N. Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. *Image Vis. Comput.* **2019**, *88*, 120–131. [\[CrossRef\]](#)

18. Ahn, H.; Cho, H.-J. Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system. *Pers. Ubiquit Comput.* **2022**, *26*, 385–394. [[CrossRef](#)]
19. Ahmed, I.; Din, S.; Jeon, G.; Piccialli, F.; Fortino, G. Towards Collaborative Robotics in Top View Surveillance: A Framework for Multiple Object Tracking by Detection Using Deep Learning. *IEEE/CAA JAS* **2021**, *8*, 1253–1270. [[CrossRef](#)]
20. Hossain, S.; Lee, D. Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices. *Sensors* **2019**, *19*, 3371. [[CrossRef](#)] [[PubMed](#)]
21. Cheng, P.; Xiong, Z.; Bao, Y.; Zhuang, P.; Zhang, Y.; Blasch, E.; Chen, G. A Deep Learning-Enhanced Multi-Modal Sensing Platform for Robust Human Object Detection and Tracking in Challenging Environments. *Electronics* **2023**, *12*, 3423. [[CrossRef](#)]
22. Liu, H.; Zhang, H.; Wang, X.; Sun, R.; Zhang, H. Multimodal Perception-Based Target Tracking for Child Safety Monitoring in Home Environments. *Int. Core J. Eng.* **2025**, *11*, 65–72. [[CrossRef](#)]
23. Ye, Y.; Ma, X.; Zhou, X.; Bao, G.; Wan, W.; Cai, S. Dynamic and Real-Time Object Detection Based on Deep Learning for Home Service Robots. *Sensors* **2023**, *23*, 9482. [[CrossRef](#)] [[PubMed](#)]
24. Zha, C.; Luo, S.; Xu, X. Infrared multi-target detection and tracking in dense urban traffic scenes. *IET Image Process.* **2024**, *18*, 1613–1628. [[CrossRef](#)]
25. Chung, M.-A.; Chai, S.-Y.; Hsieh, M.-C.; Lin, C.-W.; Chen, K.-X.; Huang, S.-J.; Zhang, J.-H. YOLO-LSD: A Lightweight Object Detection Model for Small Targets at Long Distances to Secure Pedestrian Safety. *IEEE Access* **2025**, *13*, 83061–83070. [[CrossRef](#)]
26. Villegas, W.E.; Sánchez-Viteri, S.; Luján-Mora, S. Real-Time Recognition and Tracking in Urban Spaces Through Deep Learning: A Case Study. *IEEE Access* **2024**, *12*, 95599–95612. [[CrossRef](#)]
27. Lin, J.-J.; Guo, J.-I.; Shivanna, V.M.; Chang, S.-Y. Deep Learning Derived Object Detection and Tracking Technology Based on Sensor Fusion of Millimeter-Wave Radar/Video and Its Application on Embedded Systems. *Sensors* **2023**, *23*, 2746. [[CrossRef](#)] [[PubMed](#)]
28. Chen, M.; Banitaan, S.; Maleki, M. Enhancing Pedestrian Group Detection and Tracking Through Zone-Based Clustering. *IEEE Access* **2023**, *11*, 132162–132179. [[CrossRef](#)]
29. Lecrosnier, L.; Khemmar, R.; Ragot, N.; Decoux, B.; Rossi, R.; Kefi, N.; Ertaud, J.-Y. Deep Learning-Based Object Detection, Localisation and Tracking for Smart Wheelchair Healthcare Mobility. *Int. J. Environ. Res. Public Health* **2021**, *18*, 91. [[CrossRef](#)] [[PubMed](#)]
30. Cao, H.; Cheng, L.; Shen, Z.; Huang, C.; Huang, H.; Wang, F.-Y. Detecting and Tracking 6-DoF Motion of Unknown Dynamic Objects in Industrial Environments Using Stereo Visual Sensing. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 7558–7570. [[CrossRef](#)]
31. Liu, S.; Wang, S.; Liu, X.; Lin, C.-T.; Lv, Z. Fuzzy Detection Aided Real-Time and Robust Visual Tracking Under Complex Environments. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 90–102. [[CrossRef](#)]
32. Li, F.; Zhang, R.; You, F. Fast pedestrian detection and dynamic tracking for intelligent vehicles within V2V cooperative environment. *IET Image Process.* **2017**, *11*, 833–840. [[CrossRef](#)]
33. Tang, W.; He, F.; Bashir, A.K.; Shao, X.; Cheng, Y.; Yu, K. A remote sensing image rotation object detection approach for real-time environmental monitoring. *Sustain. Energy Technol. Assess.* **2023**, *57*, 103270. [[CrossRef](#)]
34. Ai, Y.; Rui, T.; Yang, X.; He, J.; Fu, L.; Li, J.; Lu, M. Visual SLAM in dynamic environments based on object detection. *Def. Technol.* **2021**, *17*, 1712–1721. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.