DeepSORT-Gated-Deformable Tracker: Hybrid Deep Association and Deformable Feature Fusion for Robust Multi-Object Tracking

Taksh Mandar, Parth Saraswat, Ashish Kumar
Department of Computer Science, Netaji Subhas University of Technology (NSUT), India

***Abstract*—Multi object tracking (MOT) in complex scenarios such as crowded video streams remains a significant challenge due to frequent occlusions and non-rigid deformations. There are some popular tracking by detection framework like DeepSort [1] which combines motion based Kalman filtering [2] with an appearance descriptor but their performance degrades very much when the object appearance changes drastically. The standard rigid appearacne features fail to maintain a consistent identity which leads to high identity switches. We propose a new deformation aware tracking framework that enchances appearance matching part of online tracking. We propose Deformable Deep Sort(DDSORT) which integrates (1)Kalman Filter based motion prediction, (2)Deep appearance embeddings [3], (3)Deformable Convolution [4] for deformable features integration and (4) a gated fusion module based on [5] that is used to balance between deformable and standard features. The proposed mechanism enhances the robustness of DeepSort algorithm [1] as it uses deforamble feautres for objects which are more reliable.**

***Index Terms*—Multi-object tracking, deformable convolution, gated fusion, deep association metric, deep learning.**

I. Introduction

MULTI-OBJECT tracking (MOT) is a fundamental task in computer vision which has critical applications in autonomous driving, robotic navigation and intelligent surveillance. The primary goal of MOT is to detect objects of interests in video sequnces and maintain their unique identities and establish trajectories/tracks over time.

In recent years , the tracking by detection framework has become the dominant approach for MOT due to remarkable success of deep learning based object detection (YOLO [6], Faster RCNN [7]). In this framework , detections are first generated for each frame independently and data association is then performed to link these detections over time to form trajectories.

SORT [8] and DeepSORT [1] represent the two most influential and widely adpoted MOT frameworks due to their simplicity and real time performance. SORT employs a Kalman filter [2] for motion prediction and the Hungarian algorithm [9]for assginment , while DeepSORT enhances robustness by incorporating a deep appearance embedding for re-identification(Re-ID) [3].

However, the appearance model in DeepSORT and similar trackers is based on standard CNNs, which extract features using a rigid, fixed-grid sampling mechanism.This design is flawed in the sense that when tracking objects which undergo a lot of deformations, they are unable to align with the deformable object regions. This often leads to unreliable readings which leads to frequent identity switches and missed associations.

To overcome this limitation, we propose a new tracking framework **DDSORT** that uses a deformation-aware appearance representation. Our core contribution is the replacement of the rigid feature extractor with a dynamic, deformable feature extractor.

Our model has an enhanced descriptor which consists of:

- A **standard feature branch** to capture static, identity-preserving cues.
- A **deformable convolution branch** that captures features invariant to geometric transformations and deformations.
- A **gated fusion module** that intelligently weighs and combines the feature maps from both branches (standard and deformable), producing a single robust descriptor resilient to deformations and occlusions.

.

This descriptor's feature vector is then integrated into DeepSORT data association framework, combining its robustness with the effectively proven Kalman filter used for motion estimation.

II. Related Work

Multiple Object Tracking (MOT) has evolved significantly with advances in detection, feature representation, and data association. The current state of the art MOT trackers can be broadly classified into several categories, including tracking-by-detection frameworks [1], [8], deformable feature learning methods [4], [5], Siamese-network-based approaches [10]–[12], and deep appearance modelling with re-identification [3], [13], [14].

A. Tracking by detection frameworks

Traditional online MOT approaches, including SORT [8] and DeepSORT [1], rely solely on motion estimation via the Kalman filter [2] and frame-wise association using the Hungarian algorithm [9]. DeepSORT provides a significant improvement by introducing a deep appearance embedding for re-identification (Re-ID) [3], [13], enabling it to handle occlusions and substantially reduce identity switches in crowded scenes. However, a key limitation persists: appearance embeddings in DeepSORT are extracted using rigid CNN filters that are trained on person Re-ID datasets, making them

highly sensitive to variations in object shape, pose, perspective distortion, and non-rigid deformations. Such variations are common in real-world MOT applications involving humans, animals, articulated objects, or sports players, and fixed-grid convolution often fails to extract stable, identity-preserving features [4].
This motivates the need for deformation-aware appearance models that can capture both static and non-rigid structural variations during tracking [5].

B. Deformable and Gated Feature Learning

To address the limitations of rigid feature extraction, a separate line of research has focused on creating representations that are immune to geometric transformations. **Deformable convolution** was introduced in [4] to overcome this limitation. It extends standard convolutional layers by learning 2D offsets for each sampling location in the image, allowing the receptive field to deform dynamically and adapt to the object's geometry and pose. This mechanism enables the network to learn features that are inherently more robust to geometric variations.

While deformable convolution is effective at capturing variant shapes and structures, it can sometimes neglect fine-grained, static texture and color cues that are also critical for re-identification. The Gated Deformable Tracking (GDT) model [5] addresses this by introducing a gated fusion mechanism that combines features from both the standard and deformable branches. This mechanism achieves superior tracking performance in single-object scenarios involving significant deformation by balancing geometric adaptability and appearance fidelity.

C. Siamese Network based approaches

Siamese network based approaches have gained significant traction in the visual tracking community due to their favourable trade-off between accuracy and computational efficiency [10]–[12]. Unlike traditional tracking-by-detection frameworks, these methods formulate tracking as a similarity-learning problem, where a ground-truth template of the target is compared against multiple candidate patches within a search region, and the patch with the highest similarity score is selected as the new target location. A fully convolutional Siamese architecture [10] is typically used to compute similarity by processing paired inputs and producing a confidence map or similarity score.

Since these models generally require minimal or no online training, they operate at real-time speeds and are less susceptible to drifting caused by continuous model updates. This also makes them computationally efficient. However, achieving state-of-the-art performance with Siamese trackers depends heavily on extensive offline training using large-scale datasets [3], [13].

D. Deep Appearance Modelling and Re-Identification

In DeepSORT [1], appearance modelling plays a central role in modern data association. Classical approaches such as SORT [8] rely solely on geometric overlap or motion-distance metrics, which are effective only when the state uncertainty is low and occlusions are short. However, in crowded scenes, partial occlusions or camera motion, geometric information alone becomes insufficient for reliable data association. This limitation motivated the incorporation of deep appearance embeddings and re-identification (Re-ID) [3], [13].

The main aim of the appearance model is to learn an embedding function that maps detections into a normalized feature space where cosine distance is used to measure similarity. The CNN used in DeepSORT is trained on large-scale Re-ID datasets such as MARS [3] and Market-1501 [13] to maximize intra-class compactness and inter-class dispersion. This is typically achieved through metric learning approaches such as triplet loss or softmax-based embedding learning [14], enabling the model to produce discriminative appearance vectors even under substantial viewpoint and illumination changes.

III. Proposed Methodology

This section presents the complete workflow of the proposed **Deformation-Aware Deep SORT (DDSORT)** framework. The design extends the classical DeepSORT pipeline [1] by integrating a gated deformable feature extractor (GDT) inspired by deformable convolutional networks [4] and the gated fusion mechanism introduced in [5]. This provides robustness to geometric variations such as pose changes, deformation, and non-rigid motion. The full pipeline contains four main components: object detection (using YOLOv7 [6]), Kalman filter–based motion estimation [2], GDT-based appearance extraction [5], and motion–appearance data association using the Hungarian algorithm [9]. The framework operates in two stages: initialization on the first frame and online tracking for subsequent frames.

A. Initialization Stage (Frame 1)

1) Object Detection: Given the first frame I_1, a pre-trained detector (we used YOLOv7 [6]) is applied to produce a detection set

$$\mathcal{D}_1 = \{d_1^{(1)}, d_2^{(1)}, \ldots, d_N^{(1)}\}. \tag{1}$$

Each detection supplies the bounding box (x, y, w, h) used as the measurement input for the tracker.

2) GDT Feature Extraction: Each detection is cropped and passed through the GDT module, which contains a standard CNN branch, a deformable convolution branch, and a gating module.

- **Deformable Convolutional Module:** Following the formulation in [4], we employ deformable convolution layers to adapt sampling locations to object shape and pose variations. As illustrated in Fig. 2, these layers learn a set of 2D offsets that modify the sampling grid, producing a deformation-aware feature map F_{deform}.
- **Gated Fusion Mechanism:** A parallel standard convolution branch extracts the normal appearance feature map F_{standard}. To intelligently combine standard and deformable features, a gating module computes a spatial

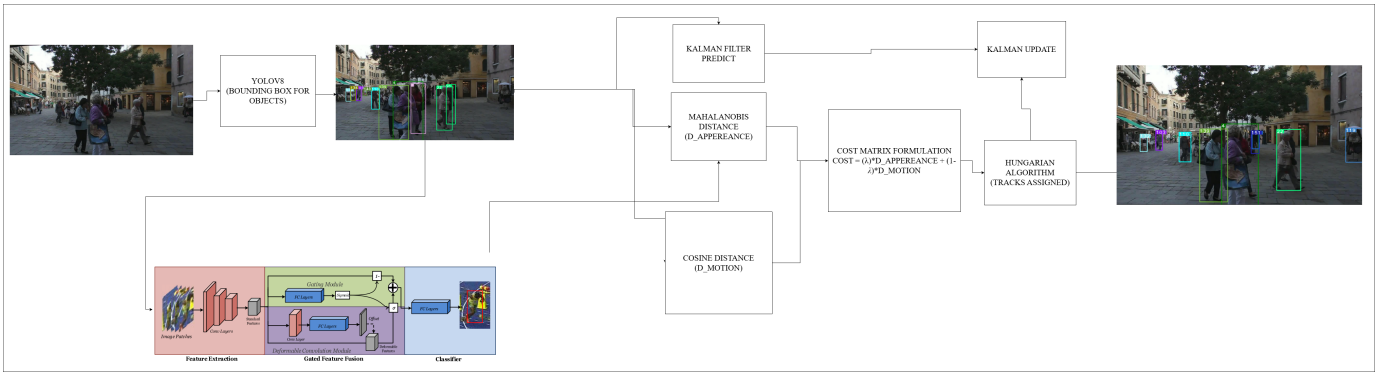

Fig. 1. System overview of the proposed Deformation-Aware DeepSORT (DDSORT) tracking pipeline. The architecture consists of four core stages: (1) input and feature extraction using YOLOv7 and the GDT module, (2) motion prediction via a Kalman filter, (3) data association using fused motion and appearance cues with the Hungarian algorithm, and (4) generation of final tracked object trajectories.

gate G that adaptively controls their relative contribution, as proposed in the GDT model [5]:

$$G = \sigma(W_g * [F_{\text{standard}}, F_{\text{defor}}] + b_g)\,, \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid activation, W_g and b_g are learnable parameters, and $[\cdot,\cdot]$ denotes channel-wise concatenation. The final fused feature representation is given by

$$F_{\text{fused}} = G \odot F_{\text{deform}} + (1 - G) \odot F_{\text{standard}}, \tag{3}$$

where $\odot$ denotes element-wise multiplication.

The fused feature map is then passed through an average global pooling layer to produce the final k-dimensional deformation-aware descriptor for $d_j^{(1)}$:

$$L_j^{(1)} = \text{GDT}(d_j^{(1)}). \tag{4}$$

3) Track Initialization: Each detection initializes a new track T_j. We follow the state-space definition from [1], creating a track T_j with:

- A standard Kalman filter KF_j which models linear constant-velocity motion [2].
- An 8-dimensional state vector $x = (u, v, \gamma, h, \dot{u}, \dot{v}, \dot{\gamma}, \dot{h})$, where (u, v) is the bounding-box center, γ is the aspect ratio, h is the height, and the dotted terms denote the respective velocities (assumed constant under linear motion).
- A new appearance gallery $\mathcal{G}_j = \{L_j^{(1)}\}$, which stores the extracted descriptor.
- An age counter initialized to 0 (i.e., Age $= 0$).

The active track set at the end of initialization is

$$\mathcal{T}_1 = \{T_1, \ldots, T_N\}. \tag{5}$$

B. Per-Frame Tracking (For Frames $t > 1$)

Each subsequent frame undergoes four major steps: motion prediction (using the Kalman Filter [2]), feature extraction, data association, and track management.

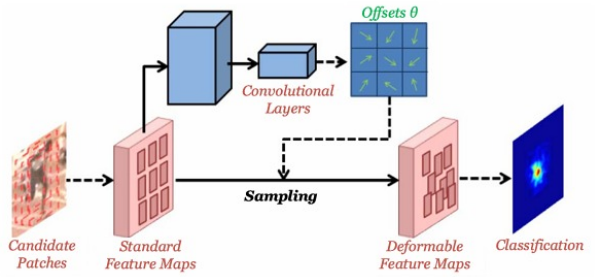

Fig. 2. Overview of the proposed GDT feature extraction module. The deformable sampling offsets are learned as proposed in [5]. Candidate image patches are processed through standard and deformable convolution branches, where the learned offsets modify the sampling grid to produce deformation-aware feature maps. These maps are then fused by a gating mechanism to generate the final robust appearance representation.

1) Motion Prediction: For each active track $T_i \in \mathcal{T}_{t-1}$, the associated Kalman Filter KF_i predicts the track's state in the current frame t before observing new detections. The Kalman filter predicted next state is given by:

$$\hat{X}_i^{(t)} = KF_i.\text{predict}(). \tag{6}$$

2) New Detection and GDT Feature Extraction: The object detector (YOLOv7 [6]) is executed on frame I_t, which produces a new set of detections for the current frame as follows:

$$\mathcal{D}_t = \{d_1^{(t)}, \ldots, d_M^{(t)}\}, \tag{7}$$

and as in the initialization stage, each detection $d_j^{(t)}$ is processed by the GDT module [5] to obtain its deformation-aware features:

$$L_j^{(t)} = \text{GDT}(d_j^{(t)}). \tag{8}$$

C. Data Association

The association problem is to match the set of new detections $\mathcal{D}_t$ with the set of predicted tracks $\mathcal{T}_{t-1}$, following the tracking-by-detection paradigm used in SORT [8] and DeepSORT [1].

Algorithm 1 DD-SORT Tracking Pipeline

Input: Video frames $\{I_t\}$, detector $\mathcal{F}_{det}$, GDT extractor $\mathcal{F}_{GDT}$, Kalman filters, gating thresholds t_1, t_2, weight λ

Initialization (Frame 1)
Detect objects: $\mathcal{D}_1 = \mathcal{F}_{det}(I_1)$
for each detection $d_j^{(1)}$ **do**
    Extract descriptor $L_j^{(1)} = \mathcal{F}_{GDT}(d_j^{(1)})$
    Initialize track T_j with KF_j, state vector, gallery $\{L_j^{(1)}\}$, Age = 0
end for
For each next frame $t = 2, \ldots, T$
1) Motion Prediction
for each track T_i **do**
    Predict state $\hat{X}_i^{(t)} = KF_i$.predict()
end for
2) Detection & Appearance Extraction
$\mathcal{D}_t = \mathcal{F}_{det}(I_t)$
for each detection $d_j^{(t)}$ **do**
    $L_j^{(t)} = \mathcal{F}_{GDT}(d_j^{(t)})$
end for
3) Motion & Appearance Cost Computation
for each track i and detection j **do**
    Compute motion distance and gate $b_{i,j}^{(1)}$
    Compute appearance distance and gate $b_{i,j}^{(2)}$
    Combined cost $C_{i,j} = \lambda d_{\text{motion}}(i,j) + (1-\lambda) d_{\text{appearance}}(i,j)$
end for
4) Matching Cascade + Hungarian Assignment
Output matched pairs, unmatched tracks, unmatched detections
5) Track Management
for each matched pair **do**
    Update KF and descriptor gallery; reset Age
end for
for each unmatched track **do**
    Age++; remove if Age $>$ MaxAge
end for
for each unmatched detection **do**
    Initialize new tentative track
end for
return Updated active tracks $\mathcal{T}_t$

1) Motion Cost: The Mahalanobis distance, as used in DeepSORT [1], is computed as

$$d_{\text{motion}}(i,j) = (d_j^{(t)} - \hat{y}_i)^\top S_i^{-1} (d_j^{(t)} - \hat{y}_i), \tag{9}$$

followed by a gating operation based on the Kalman filter covariance [2] using $b_{i,j}^{(1)}$.

2) Appearance Cost: Appearance matching follows the deep appearance metric used in DeepSORT [1], where embeddings are learned from large-scale Re-ID datasets such as MARS [3] and Market-1501 [13]. The appearance distance is computed as

$$d_{\text{appearance}}(i,j) = \min_{L_{i,k} \in \mathcal{G}_i} \left(1 - (L_j^{(t)})^\top L_{i,k}\right), \tag{10}$$

with its corresponding appearance gate $b_{i,j}^{(2)}$ applied to filter unlikely associations.

3) Combined Cost: The fused distance metric follows the motion–appearance association strategy used in DeepSORT [1], where motion and appearance cues are combined to improve robustness. The fused cost is defined as

$$C_{i,j} = \lambda d_{\text{motion}}(i,j) + (1-\lambda) d_{\text{appearance}}(i,j), \tag{11}$$

and the final admissibility constraint is given by

$$b_{i,j} = b_{i,j}^{(1)}\, b_{i,j}^{(2)}, \tag{12}$$

which ensures that both motion-based gating [2] and appearance-based gating [1] are satisfied.

D. Track Management

Matched tracks are updated with the new measurement, their descriptors are added to the appearance gallery, and Age is reset, following the update rules defined in DeepSORT [1]. Unmatched tracks age incrementally and are removed when exceeding the maximum allowed age, as in the original SORT/DeepSORT framework [1], [8]. New unmatched detections spawn tentative tracks, which are confirmed only after sufficient consecutive associations, following the same initialization strategy proposed in [1].

E. Final Active Track Set

The active set of tracks at time t is

$$\mathcal{T}_t = \{T_1, \ldots, T_{N_t}\}, \tag{13}$$

which is passed to the next iteration as done in standard MOT pipelines such as SORT [8] and DeepSORT [1].

IV. Experiments

A. Implementation Details and Experimental setup

1) GDT Architecture: The GDT feature extracter consists of 3 main parts , first one is the standard feature extractor for that we have used the first 16 layers of VGG-16 model [15], the next part consists of the Deformable features it consists of a convolution layer and a fully connected layer which generates the output of 3 x 3 x 2 [4]. Then the feature map is constructed using the bilinear sampler [16]. The final part is the gating module which consists of two consecutive fully connected layers followed by a sigmoid activation and outputs a 3 × 3 gating values [5]. After all these layers the the gating values are applied on the standard features and the deformable features to get the final feature descriptor, which is then used in the proposed framework.

2) Training GDT: The GDT module is trained in a 3 step process but for it to train we also add a classifier to make the training feasible, so that the model learns to discriminate images and in the end is able to give a good descriptor of it. The dataset used is GOT-10K [24] and DeformSORT. So first only the (base (standard feature) + classifier) is trained for 75k iterations, then (base + deform + classifier) for 75K iterations and then (base + deform + gate + classifier) for the last 60K iterations. So in total approximately it was trained

		MOTA ↑	MOTP ↑	MT ↑	ML ↓	ID ↓	FM ↓	FP ↓	FN ↓
KDNT [17]*	BATCH	68.2	79.4	41.0%	19.0%	933	1093	11479	45605
LMP-p [18]*	BATCH	**71.0**	**80.2**	**46.9%**	21.9%	434	**587**	7880	**44564**
MCMOT_HDM [19]	BATCH	62.4	78.3	31.5%	24.2%	1394	1318	9855	57257
NOMTwSDP16 [20]	BATCH	62.2	79.6	32.5%	31.1%	**406**	642	**5119**	63352
EAMTT [21]	ONLINE	52.5	78.8	19.0%	34.9%	910	**1321**	4407	81223
POI [22]*	ONLINE	**66.1**	79.5	34.0%	20.8%	805	3093	5061	**55914**
SORT [8]*	ONLINE	59.8	79.6	25.4%	22.7%	1423	1835	8698	63245
Deep SORT* [1]	ONLINE	61.4	79.1	32.8%	**18.2%**	781	2008	12852	56668
DDSORT (Ours)*	ONLINE	49.2	75.2	31.4%	30.4%	**492**	2402	13024	62034

TABLE I
TRACKING RESULTS ON THE MOT16 [23] CHALLENGE. WE COMPARE TO OTHER PUBLISHED METHODS WITH NON-STANDARD DETECTIONS. METHODS MARKED WITH * USE DETECTIONS PROVIDED BY [22].

for 200K iterations. The machine used was Kaggle's notebook which provided the P100 GPU [25]. At the end the final feature descriptor is ready.

B. Results

Metric Terminologies:

- **Multi-object tracking accuracy (MOTA):** Summary of overall tracking accuracy in terms of false positives, false negatives and identity switches [26].
- **Multi-object tracking precision (MOTP):** Summary of overall tracking precision in terms of bounding box overlap between ground-truth and reported location [26].
- **Mostly tracked (MT):** Percentage of ground-truth tracks that have the same label for at least 80% of their life span [26].
- **Mostly lost (ML):** Percentage of ground-truth tracks that are tracked for at most 20% of their life span [26].
- **Identity switches (ID):** Number of times the reported identity of a ground-truth track changes [26].
- **Fragmentation (FM):** Number of times a track is interrupted by a missing detection [26].

MOT-16: The MOT-16 dataset, introduced as part of the MOTChallenge benchmark [23], is a standard and widely used dataset for evaluating Multiple Object Tracking (MOT) algorithms. Its primary goal is to provide a standardized framework for the fair and objective comparison of tracking methods [23]. We evaluated on the train set of the MOT-16, which consisted of 7 difficult sequences. The value of $\lambda = 0$ was set for evaluation, i.e., the cosine distance is used, but the Mahalanobis distance is not completely discarded as it is used to disregard infeasible assignments based on the possible location predicted by the Kalman filter [2].

We have compared it to multiple tracking algorithms available in the literature, including KDNT [17], EAMTT [21], SORT [8] , and Deep SORT [1]. The number of ReIDs, as predicted, has gone down but at the cost of MOTA and FN. The ID has reduced from 781 to 492, which is a significant improvement [26]

V. CONCLUSION

We introduced **DDSORT**, a deformable deep association tracking framework that fuses DeepSORT's [1] robust identity modeling with GDT's adaptive deformable convolution and gated fusion [5]. The method achieves low ID and comparable MOTA and other metrics on similar/lower runtime. The lower ID suggests that using the GDT feature extractor for multiple object tracking indicates stable and consistent tracking with lower identity switches.

TABLE II
GDT ARCHITECTURE [4], [5], [15], [16]

Module	Key Components / Layers	Output / Purpose
Feature Extractor	First 16 Layers of VGG-16 [15]	Produces the standard feature map (X)
Deformable Conv.	- 1x Convolutional Layer [4] - 1x Fully Connected Layer [4] - 1x Bilinear Sampler [16]	- Generates $3 \times 3 \times 2$ offsets - Reconstructs deformable map (X')
Gating Module	- 2x Consecutive FC Layers [5] - 1x Sigmoid Activation [5]	- Outputs 3×3 gating values (σ)

Future Scope

The future work that can be done in this tracking framework is introducing a more robust feature descriptor by increasing the complexity of the model thus leading to improvement in the MOTA and also reduce the number of FN. Instead of kalman filter which is linear filter, non-linear filters like extended or unscented Kalman filters [27], [28] could be used to improve performance in such motion. We would also suggest testing it on newer datasets such as DanceTrack [29], SportsMOT [30] and AnimalTrack [31].

REFERENCES

[1] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[2] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, 1995.

[3] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su *et al.*, "Mars: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 868–884.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 764–773.

[5] W. Liu, Y. Song, D. Chen, S. He, Y. Yu, T. Yan, G. Hancke, and R. W. H. Lau, "Deformable object tracking with gated fusion," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3766–3779, 2019.

[6] 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," *arXiv preprint arXiv:2207.02696*, 2023.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[9] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision (ECCV)*, 2016.

[11] B. Li, J. Yan, W. Wu, and Z. Zhu, "High performance visual tracking with siamese region proposal network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] Q. Wang, Z. Teng, J. Xing, J. Gao, and W. Hu, "Learning attentions: Residual attentional siamese network for high-performance online visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] L. Zheng, L. Shen, Y. Tian, J. Wang, S. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[17] K. Raja *et al.*, "Kalman deep network tracker," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017.

[18] S. Tang, M. Andriluka, B. Andres, S. Roth, and B. Schiele, "Learning multi-cue representation for multi-object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Multi-cue and multi-object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] W. Choi, "Near-online multi-target tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Eamtt: Evolving appearance model for multi-target tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2016.

[22] Q. Yu, F. Poiesi, and A. Cavallaro, "Person of interest: Re-identification-based multi-object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016.

[23] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," in *arXiv preprint arXiv:1603.00831*, 2016.

[24] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] "Kaggle gpu compute service," https://www.kaggle.com, 2024.

[26] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[27] S. Julier and J. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing*, 1997.

[28] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[29] Y. Wang, J. Xing *et al.*, "Dancetrack: Multi-object tracking in uniform appearance and diverse motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[30] P. Xu *et al.*, "Sportsmot: A multi-object tracking benchmark for sports understanding," in *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[31] A. Gupta *et al.*, "Animaltrack: A benchmark for multi-animal tracking in the wild," in *European Conference on Computer Vision (ECCV)*, 2022.