IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 July 2022, accepted 3 August 2022, date of publication 8 August 2022, date of current version 11 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197157

RESEARCH ARTICLE

LMOT: Efficient Light-Weight Detection and Tracking in Crowds

RANA MOSTAFA, HODA BARAKA, AND ABDELMONIEM BAYOUMI, (Member, IEEE)
Department of Computer Engineering, Faculty of Engineering, Cairo University, Giza 12613, Egypt
Corresponding author: AbdElMoniem Bayoumi (abayoumi@cu.edu.eg)

ABSTRACT Multi-object tracking is a vital component in various robotics and computer vision applications. However, existing multi-object tracking techniques trade off computation runtime for tracking accuracy leading to challenges in deploying such pipelines in real-time applications. This paper introduces a novel real-time model, LMOT, i.e., Light-weight Multi-Object Tracker, that performs joint pedestrian detection and tracking. LMOT introduces a simplified DLA-34 encoder network to extract detection features for the current image that are computationally efficient. Furthermore, we generate efficient tracking features using a linear transformer for the prior image frame and its corresponding detection heatmap. After that, LMOT fuses both detection and tracking feature maps in a multi-layer scheme and performs a two-stage online data association relying on the Kalman filter to generate tracklets. We evaluated our model on the challenging real-world MOT16/17/20 datasets, showing LMOT significantly outperforms the state-of-the-art trackers concerning runtime while maintaining high robustness. LMOT is approximately ten times faster than state-of-the-art trackers while being only 3.8% behind in performance accuracy on average leading to a much computationally lighter model.

INDEX TERMS Multi-object tracking, pedestrian tracking, joint detection and tracking, object detection, deep learning.

I. INTRODUCTION

Multi-Object Tracking (MOT) is an inevitable feature in modern autonomous robotics systems. Estimating trajectories for objects of interest through time is crucial for a wide range of real-world applications. Such applications may range from indoor mobile service robots that need to navigate in a human-friendly manner to autonomous vehicles that track nearby vehicles and pedestrians. Therefore, such real-world applications require a robust multi-object tracking approach that satisfies real-time compatibility [1], [2].

A key challenge for multi-object tracking is the trade-off between achieving high accuracy and real-time compatibility. Current top-performing trackers follow mainly two paradigms: tracking-by-detection and joint-detection-and-tracking. Tracking-by-detection approaches decouple the detection and tracking tasks via introducing a tailored model for each of them, leading to robust multi-object tracking. However, they suffer from computational inefficiency due to having two separate models that run sequentially. On the other hand, joint-detection-and-tracking approaches extract detection and tracking features using a single model for better computational efficiency.

This paper introduces a novel real-time and robust joint-detection-and-tracking model, entitled LMOT, i.e., Light-weight Multi-Object Tracker. LMOT simplifies the DLA-34 encoder [3] and combines it with a linear transformer to track pedestrians in crowded environments. DLA-34 is commonly used by the state-of-the-art trackers [4], [5], [6], [7] to extract essential spatial and semantic information for detection; however, it is computationally expensive and does not fit real-time applications. Thus, we introduce a simplified version of Deep Layer Aggregation Network (DLA-34) to ensure computational efficiency. Additionally, to avoid accuracy degradation, we support the simplified DLA-34 with a computationally efficient linear transformer that generates tracking features. After that, we fuse the generated detection and tracking features via our proposed multi-layer

The associate editor coordinating the review of this manuscript and approving it for publication was Janmenjoy Nayak.

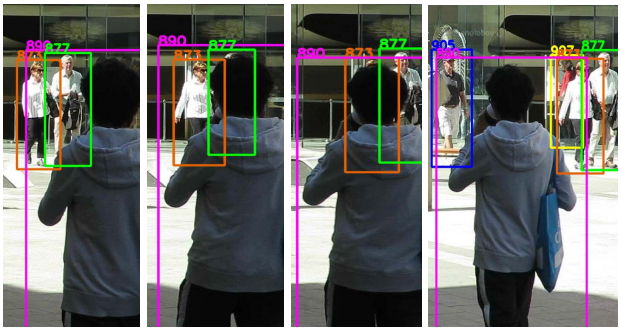

FIGURE 1. An example from the MOT17 dataset showing the robustness of LMOT and its ability to estimate trajectories of fully occluded objects.

fusion module into a single feature map for joint-detection-and-tracking. Furthermore, we introduce a novel enhanced two-stage online data association module that applies Kalman filter [8] to estimate tracklets' locations leading to tracklets that are robust to challenging real-world scenarios.

We evaluated our model on the MOT16/17/20 datasets, which are well-known benchmarks used by state-of-the-art trackers. These datasets include challenging real-world scenarios for tracking pedestrians in crowds, occlusions, illumination, and camera motion and orientation challenges. Such real-world experiments show the robustness of LMOT and its ability to track objects of interest and infer their location even when they are fully occluded, as demonstrated in Figure 1.

To conclude, the contributions of this paper are as follows:

1) Presenting a novel real-time and robust multi-object tracking model that achieves a significant computational runtime improvement compared to the state-of-the-art.
2) Exploring the possibility of simplifying the DLA-34 encoder to generate powerful detection features that are computationally efficient.
3) Generating tracking features using a linear transformer to avoid accuracy degradation caused by the simplified DLA-34.
4) Fusing detection and tracking features in a unified feature map using a multi-stage scheme.

Additionally, we organize the rest of this paper as follows; Section II explores the related work to multi-object tracking and the existing paradigms. Next, Section III discusses our proposed model, LMOT, in detail. Then, Section IV demonstrates our real-world experiments and discusses the achieved results, besides showing some insightful ablation studies.[1] Finally, Section V concludes our findings, highlights our model's limitations and proposes future research directions to address such limitations.

II. RELATED WORK

Early tracking methodologies operate offline and perform global optimization for the whole sequence of frames in a fast and simple way [9], [10], [11], [12]. However, such early methodologies tend to suffer when tackling real-world scenarios, requiring complex features to represent objects of interest. Thus, the current research trend of multi-object tracking exploits the deep learning techniques advancements and follows two paradigms: tracking-by-detection and joint-detection-and-tracking.

Tracking-by-detection approaches [13], [14], [15], [16] separate detection and tracking steps. These approaches deal only with the current frame and usually apply popular off-shelf detector networks to generate detection bounding boxes for objects of interest, such as Deformable Part Model (DPM) [17], Faster Recurrent Convolutional Neural Networks (Faster-RCNN) [18] and Scale Dependent Pooling (SDP) [19]. Then, they use separate tracking models to link the detected objects over time. For instance, SORT [13] applies Faster-RCNN [18] for detection and then employs a Kalman filter [8] for tracking. Afterward, DeepSORT [14] extends the SORT [13] approach by extracting distinct features after detecting the objects of interest to help in their re-identification through time. Moreover, Bae [15] proposed a tracking framework that models objects' visual and radar features and their affinity using a confidence-based data association model and a visual learning object model. Alternatively, Liang *et al.* [16] use graph neighbor networks to model full contextual relations for each tracklet with its surrounding neighbor tracklets for effective data association. Accordingly, the tracking-by-detection paradigm's powerful aspect is that for each task, we build the most convenient model for its goal. However, due to independent models, this paradigm is computationally expensive to train and slow concerning runtime execution.

Joint-detection-and-tracking paradigm adopts a unified framework that serves both detection and tracking. For example, CTracker [20] chains adjacent bounding boxes across frames to generate tracklets from consecutive frames. Furthermore, TubeTK [21] links all sequence frames through time to represent tracklets' moving trajectories to attain more global representations. Alternatively, JDE [22] generates distinctive features at objects' centers to incorporate them with data association and re-identification. Moreover, CenterTrack [4] extends CenterNet [3] using its deep layer aggregation architecture, i.e., DLA-34, to represent objects using their center points and track them based on their offsets in previous frames. Similarly, TraDeS [5] uses center points to represent objects with DLA-34 backbone and generates tracking features using a cost volume association module.

The state-of-the-art approaches consist of GSDT [6], FairMOT [7] and Trackformer [25]. GSDT relies on DLA-34 [3] as a backbone and applies a graph neural network to model relations between objects of interest. However, graph networks are computationally expensive and require much training. On the other hand, FairMOT [7] uses center points with a DLA-34 backbone and generates center re-identification embeddings for data association. However, in joint training, the re-identification loss conflicts with the detection loss, which harms the detection

[1] We made our code publicly available and the source code for the developed models and experiments can be found through: https://github.com/RanaMostafaAbdElMohsen/LMOT

TABLE 1. Summary of the recent works in the literature.

Tracker	Brief Methodology	Highlights	Limitations
Martin *et al.* [9]	multi-modal approach using Gaussian probability distribution	simple, relies on probabilistic and deterministic algorithms	offline, batch method, non-robust in complex scenarios
Berclaz *et al.* [10]	employs the K-shortest paths algorithm for data association	simple, no use of greedy algorithms that miss global minimum	offline, batch method
Milan *et al.* [11]	formulates all tracklets' locations and motions with physical constraints as an energy function	applies differentiation to reach function minimum	offline, batch method, non-robust in complex scenarios
OpenPTrack [12]	open-source, single tracker framework using RGB-D cameras	single tracking model, multi-input sensors	batch, offline method
SORT [13]	applies Faster-RCNN [18] for detections and Kalman filter [8] for association	simple, fast approach	separate detection and tracking models, Faster-RCNN [18] computationally expensive
DeepSORT [14]	employs a re-identification tracking network	higher tracking ability than SORT [13]	separate models needed, slower to run than SORT [13]
Bae *et al.* [15]	employs a learning object model and confidence-based data association	sensor fusion, high accuracy, low tracking model complexity	separate detection and tracking models
Liang *et al.* [16]	employs a graph neighbor network and models contextual features	uses graph networks, competitive tracking accuracy	very slow to train and run, uses two separate networks
CTracker [20]	links adjacent bounding boxes every two consecutive frames	joint network, extracts semantic features using Residual Networks (ResNet) [23] and Feature Pyramid Networks (FPN) [24]	feature pyramid networks are computationally expensive
TubeTK [21]	introduces bounding tubes for data association using Feature Pyramid Networks (FPN) [24]	one-shot, joint network, batch method	batch, offline method, feature pyramid networks are computationally expensive
JDE [22]	extracts distinctive, center and re-identification features for tracking	joint model, use re-identification features	suffers from scalability issues, runs slower in crowded scenes
CenterTrack [4]	employs DLA-34 network and uses greedy algorithms for data association	joint, center-point, anchor-free tracker	non-robust in crowded scenes
TraDeS [5]	applies cost volume association module with center, re-identification features for tracking	center-point, joint network, use re-identification features	scalability issues
FairMOT [7]	center-point approach that uses DLA-34 network and re-identification network	joint, re-identification network branch	suffers from scalability and runtime issues
GSDT [6]	applies graph neural network for data association and uses center, distinctive re-identification features	joint network, employs graph networks with re-identification features	suffers from scalability issues, slow to train and run
Trackformer [25]	employs vision transformers with the use of track queries for consecutive frames	joint network, tracking-by-attention mechanism	transformers are computationally expensive, slow, hard to converge

performance. Moreover, these embeddings suffer from both dimension scalability and runtime issues since the number of embeddings increases proportional to the number of objects; thus, FairMOT tends to be slow in crowded environments. Alternatively, Trackformer [25] uses vision transformers as a backbone network to model long-term dependencies and generate global decisions using the multi-head attention mechanism. It employs a tracking-by-attention mechanism using track queries for consecutive frames. However, transformers are computationally expensive, slow to train, and hard to converge. We summarize the recent works in Table 1.

As opposed to the state-of-the-art, this paper introduces a real-time multi-object tracking framework that surpasses all state-of-the-arts from the runtime perspective while achieving a very close and comparable accuracy compared to them. First, LMOT simplifies the DLA-34 network [3] and combines it with a computationally efficient linear transformer to generate robust detection and tracking features, respectively. Then, LMOT fuses these feature maps and introduces an efficient two-stage online data association technique to track the generated detections.

III. LMOT

LMOT is a lightweight multi-object tracking approach that combines detection and tracking tasks in one model. LMOT is a key-point-based tracker that represents each object of interest by its center to avoid the common problems of tracking anchors [3]. Additionally, LMOT simplifies the DLA-34 [3] encoder to achieve real-time performance and supports it with a linear transformer module to avoid accuracy drop. Accordingly, LMOT ensures computational efficiency and generates robust features. Moreover, our proposed multi-stage fusion module uses both detection and tracking features to generate a single feature map. Afterward, the detection and tracking branches take the input of the fusion module and generate the outputs. Finally, our data association module links the outputs in an online mode, generating reliable tracklets. In this section, we discuss both

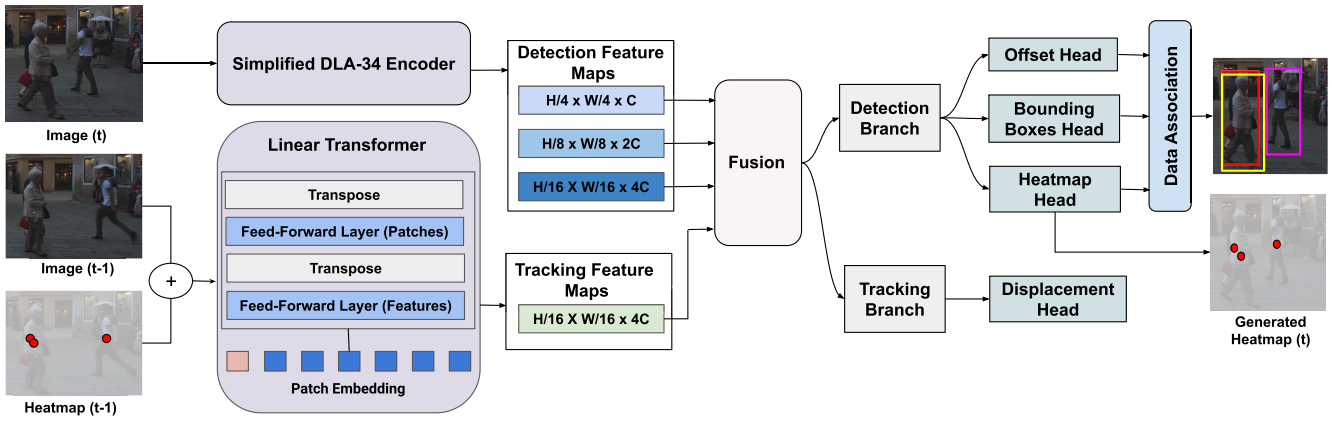

FIGURE 2. Overview of LMOT architecture. Simplified DLA-34 generates crucial detection feature maps from the current image frame encoding spatial and semantic representations. Linear transformer takes the previous image frame and its corresponding heatmap of detections' centers as an input for generating a feature map that represents tracking features. LMOT fuses tracking and detection feature maps to generate final detections and then performs data association with previous tracklets to output current tracklets.

modules' details besides our novel multi-layer fusion and data association modules.

A. SIMPLIFIED DLA-34 ENCODER

We introduce a simplified version of the Deep Layer Aggregation (DLA-34) network to improve runtime by minimizing the number of generated feature maps to account only for objects of interest to real-world trackers. The original DLA-34 [3] uses ResNet-34 [23] as a backbone to extract a broad spectrum of feature maps corresponding to five scales of the input frame, i.e., 1/2, 1/4, 1/8, 1/16, and 1/32. These feature maps represent robust semantic and spatial detection features that merge scales in a fine-to-coarse manner using iterative skip connections, besides hierarchical skip connections to preserve features from shallower layers. The goal of such scaled feature maps is to detect a wide range of objects of different sizes where some of them are too small, e.g., birds, frogs, and insects. However, such too-small objects are unlikely in real-world tracking scenarios since current real-world trackers are usually concerned with larger objects such as pedestrians and cars. Therefore, limiting the output to only the feature maps scaled as 1/4, 1/8, and 1/16 leads to a massive improvement in the runtime without much affecting the tracking performance. These selected scales keep the necessary level of details needed for detecting objects of interest while avoiding any unnecessary computational overhead. As demonstrated in Figure 2, we feed the current image frame, at time t, as an input to the simplified DLA-34 encoder to produce the three detection feature maps in a spatial and semantic representation.

B. LINEAR TRANSFORMER

In addition to the semantic and spatial detection features, LMOT also extracts tracking features to help track missed objects of interest due to occlusion or blurring. However, a crucial constraint is to perform that efficiently to realize a real-time module. Accordingly, we extend the functionality of the linear transformer of Melas-Kyriazi [26] to generate tracking features instead of performing image classification tasks in its original version. Thus, we extended the linear transformer architecture to take the heatmap of detections' centers of the previous time step, besides the corresponding image frame, as shown in Figure 2.

The linear transformer achieves a remarkably competitive accuracy in linear computational complexity since it relies on just a stack of feed-forward layers. First, we divide the concatenated input into 32 patches and perform convolutional encoding on each patch. Then, we feed the patches into a stack of six feed-forward block layers. Finally, the resultant features pass through a final normalization layer generating a 256-channel output feature map with 1/16 resolution of the original image. In addition, we note that heatmaps have identical dimensions as image frames but with only one channel representing the objects' of interest centers and concatenated to the input frame of the linear transformer.

Furthermore, it is worth mentioning that LMOT generates a heatmap of detections' centers for each time step via the heatmap head of the detection branch, as shown in Figure 2. Then, this heatmap is fed to LMOT in the consecutive time step after being scaled to fit the input dimensions. However, as for the initial frame, since there are neither prior frames nor prior detections, we feed the initial image frame and an initialized heatmap with zero values as input to the linear transformer.

C. FUSION MODULE

Our proposed multi-layer fusion technique preserves important spatial and semantic features while enhancing them with tracking features generating a single feature map for joint-detection-and-tracking. As shown in Figure 3, we fuse the tracking features obtained from the linear transformer into the multi-scale detection features of the DLA-34 in a multi-layer scheme while performing up-sampling. First, we concatenate the 256-channel feature map generated by the linear transformer module and the 256-channel map of

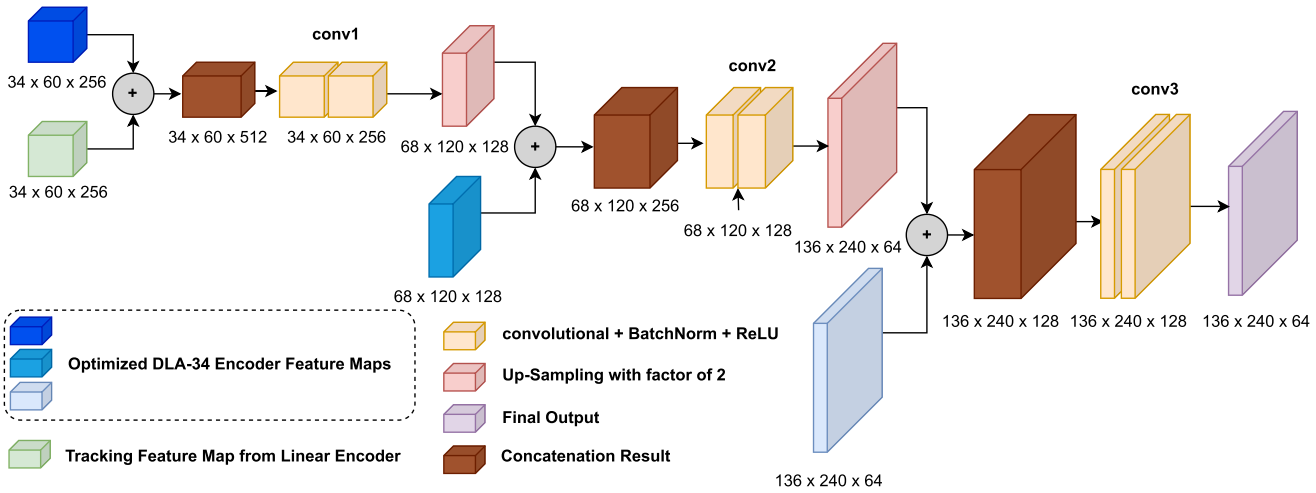

FIGURE 3. Overview of LMOT's fusion module.

the simplified DLA-34 network together. After that, we apply two convolutional layers while preserving the dimensions, and then we perform up-sampling to double the dimensions and generate a 128-channel feature map. Next, we repeat the same series of operations on the resulting feature map to fuse it with the remaining two feature maps of the simplified DLA-34 module.

Furthermore, we recover the resolution of the feature maps to quarter the input frame, which typically enhances the detection and tracking accuracy compared to lower resolutions [4]. Thus, we recover the necessary details for detection and tracking while minimizing computational overheads. Finally, it is worth mentioning that our proposed fusion technique is much lighter compared to the decoders used in GSDT [6] and FairMOT [7] since they rely on graph networks and deformable convolutions, respectively.

D. DETECTION AND TRACKING BRANCHES

We feed the feature map generated from our fusion module as an input to both our detection and tracking branches. The detection branch consists of three parallel heads: heatmap head, bounding boxes head, and offset head. They are responsible for generating a heatmap of the current frame, sizes of the bounding boxes of detected objects, and the offset of each point of each detected object relative to that object's center point, respectively. Each head consists of two 3×3 convolutional layers, each of 256-channels, followed by a 1×1 convolution layer. However, we apply a sigmoid function to the output of the heatmap head as an extra filtration step to bound its output values.

The heatmap head estimates the center points of the objects of interest in the input frame at the current time step, where potential center points correspond to regions of high intensity in the generated heatmaps. The dimensions of the heatmap correspond to quarter the resolution of the input frame to match the fusion module to preserve the necessary level of details; however, it consists of only one channel. We train this head using a pixel-wise logistic regression loss function, L_H, with focal loss [27] using the given ground-truth detected boxes, since it leads to better convergence compared to the ordinary binary cross-entropy loss [3], [4], [27], as illustrated in Eq. 1:

$$L_H = -\frac{1}{N}\sum_{xy}\begin{cases}(1-\hat{C}_{xy})^{\alpha}\log(\hat{C}_{xy}) & C_{xy}=1;\\ (1-C_{xy})^{\beta}(\hat{C}_{xy})^{\alpha}log(1-\hat{C}_{xy}) & \text{otherwise,}\end{cases} \tag{1}$$

where C_{xy} and $\hat{C}_{xy}$ are the corresponding points at the ground-truth and predicted heatmaps, respectively. Furthermore, α, β are focal loss hyper-parameters. Besides, N is the given number of objects to be detected. We compute the ground-truth heatmap by placing Gaussian-shaped peaks at the ground-truth centers provided by the training datasets, where we bound the value of each point between 0 and 1. The center points in the ground-truth heatmaps have a maximum heat intensity, i.e., a value of 1.

The bounding boxes head estimates the sizes of the detected objects' bounding boxes and stores that information in the points corresponding to those objects' centers in the generated output. Furthermore, the offset head estimates offsets of points inside the generated bounding boxes related to the detected objects' centers. Such estimated offsets help improve the localization of the detected objects. Similar to the heatmap head, each of these two heads generates an output with quarter dimensions in width and height of the input frame to match the fusion module. However, the outputs of these two heads have two channels corresponding to the width and height for the bounding boxes head and the offsets' coordinates for the offset head, respectively. Furthermore, we compute a joint loss function to train both heads, L_{SO}, as demonstrated in Eq. 2:

$$L_{SO} = \frac{1}{N}\sum_{1}^{N}\|s-\hat{s}\|_1 + \|o-\hat{o}\|_1, \tag{2}$$

where s and $\hat{s}$ correspond to given and predicted sizes of each object at any given point, respectively, while o and $\hat{o}$ correspond to their corresponding offsets, respectively.

In addition to the detection branch, we introduce a tracking branch that consists of a displacement head to estimate 2D displacements for detected objects. Such displacements correspond to the difference between objects' locations in the current frame relative to the previous frame. The main goal of this branch is to support the correct prediction of heatmaps to avoid mistakes caused by fast-moving objects via contributing to the loss function. Thus, LMOT can account for different paces of pedestrians. Furthermore, the tracking branch help in recovering from occlusions in crowded environments. Accordingly, the generated output stores the displacement information in the points corresponding to the detected objects' centers, similar to the bounding boxes head. Also, this head generates an output with quarter dimensions in width and height of the input frame while having two channels to represent the necessary details. We train this head via computing the loss of the predicted displacement compared to the given ground-truth, as explained in Eq. 3:

$$L_D = \frac{1}{N}\sum_{1}^{N} \|\hat{d}^{(t,t-1)} - d^{(t,t-1)}\|_1, \tag{3}$$

where $\hat{d}^{(t,t-1)}$ and $d^{(t,t-1)}$ are the predicted and ground-truth displacements of the object, respectively.

Finally, we compute the total loss for our model by combining the losses of the individual heads in Eq. 4 as follows:

$$L = \lambda_H * L_H + \lambda_{SO} * L_{SO} + \lambda_D * L_D, \tag{4}$$

where λ_H, λ_{SO} and λ_D are the corresponding weights of the losses.

E. ONLINE DATA ASSOCIATION

We propose an online two-stage data association method which relies on Kalman filter [8] for motion estimation. We divide data association into two stages. The first stage prioritize high-confidence detections and try associating them with all the existing tracklets in the system, including both active and lost tracklets. The lost tracklets, unlike the active ones, are those tracklets that were not associated with detections within a specific time window. In other words, we keep considering these tracklets for some future time steps to account for possible occlusions. After that, the second stage considers the low-confidence and unmatched detections that remained from the first stage and performs a second association trial with all the remaining tracklets. Finally, we initialize unmatched detections as new tracklets while keeping the unmatched tracklets for some future time steps to account for possible occlusions.

We apply a Kalman filter [8] to predict the new locations of all the tracklets before associating them to the detections. It is worth mentioning that such modification accounts for the dynamic nature of the objects of interest to improve tracking accuracy. Furthermore, we compute the intersection over union (IOU) distance between predicted tracklets' boxes and detected objects' boxes and use Jonker-Volgenant [28] algorithm for matching between the tracklets and detected objects, as opposed to the commonly used Hungarian and greedy matching algorithms.

IV. EXPERIMENTS

This section explores our real-world experiments, ablation studies, and the datasets used for training and testing. Furthermore, we discuss our evaluation metrics for comparing LMOT against state-of-the-art approaches and our experimental settings. Besides, we discuss our experimental results and our ablation studies.

A. DATASETS

This section discusses the benchmark datasets used in our experiments. Table 2 provides an overview of the pre-training and evaluation datasets used for our experiments. As for the pre-training step, it is crucial to warm up our model via exposing it to scenarios of limited crowds to allow it to learn necessary features about detecting humans to prepare it to deal with more complex scenarios with massive crowds and occlusions. Accordingly, we performed pre-training using CrowdHuman [29], CityPersons [30] and ETH [31] datasets, which provide bounding boxes for pedestrians in non-crowded scenarios.

As for training, validation, and testing, we rely upon MOT16/17/20 datasets [32], [33] that are well-known benchmarks to evaluate multi-object tracking. MOT16/17/20 datasets involve challenging real-world scenarios, besides being considered by the state-of-the-art trackers. For example, MOT16/17 datasets [32] consist of outdoor scenes with pedestrians suffering from illumination, occlusion, and camera motion challenges. Meanwhile, MOT20 [33] consists of much denser crowds with a challenging camera orientation. Therefore, we trained our pre-trained model on each of MOT16/17/20 datasets separately and evaluated our model on the corresponding testing sequences of each of them. We split the training sequences into two halves for the validation phase; however, we considered the whole training sequence for the testing phase. Furthermore, it is worth mentioning that we performed the validation on MOT17 only to avoid any bias. Additionally, we note that we do not have direct access to the ground-truths of the testing sequences of any of the MOT datasets since tests are evaluated online on the MOT datasets platform without disclosing any ground-truths.

It is worth mentioning that MOT16/17/20 datasets provide two modes of evaluation: private and public modes. The private model evaluates both the detection and tracking capabilities of the trackers. In other words, trackers must generate their detections. However, the public mode only considers the tracking capabilities via providing trackers with a set of given detections by the datasets. Accordingly, we evaluate LMOT in both modes.

FIGURE 4. Qualitative results of LMOT's multi-object-tracking performance on a test sequence from MOT16/17/20 benchmarks, respectively. Each row shows successive testing frames per benchmark with the generated tracking output boxes and identities. MOT16 sequence demonstrates blurring challenges while MOT17/20 sequences show illumination and massive crowds challenges, respectively, besides camera orientation challenges.

TABLE 2. Summary of the datasets.

Dataset	Used for Pre-training	Used for Experimental Evaluation	Highlights
CrowdHuman [29]	✓	✗	rich in human annotations, has 15,000 images, serves as a powerful pre-training dataset
CityPersons [30]	✓	✗	has 5000 high-quality images, diverse images under different scenarios and background conditions
ETH [31]	✓	✗	consists of 4200 images captured from 5 sequences
MOT16 [32]	✗	✓	consists of 14 outdoor real video sequences filled with high density of crowds and diverse scenes, suffers from illumination, occlusion, and camera motion challenges
MOT17 [32]	✗	✓	same video sequences as MOT16, supports detection mode with three different object detectors: DPM [17], Faster-RCNN [18] and SDP [19] for better performance evaluation
MOT20 [33]	✗	✓	consists of 8 sequences extracted from 3 scenes, consists of much massive, denser crowds with challenging camera orientation

B. EVALUATION METRICS

Our main goal is to achieve a fast model with satisfactory accuracy. Therefore, we evaluate the performance of LMOT compared to other approaches from both the runtime and accuracy perspectives. Accordingly, as for the computational aspect, we compute the average processing time per frame for LMOT compared to the state-of-the-art and all top-performing trackers. Furthermore, it is worth mentioning that we have performed all our experiments on the same hardware settings, i.e., a mobile Nvidia Geforce RTX 2070 GPU, to ensure a fair and unbiased comparison.

On the other hand, from the accuracy perspective, we compute both IDF1 and MOTA scores. The IDF1 score represents the ratio of correctly identified detections over the average number of ground-truth and computed detections, while MOTA stands for multi-object tracking accuracy and is computed in Eq. 5 as follows:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t GT_t}, \tag{5}$$

where t is the index of the evaluated frame, GT is the number of ground truth objects to be detected, FP denotes the number of false positives, FN is the number of false negatives.

Furthermore, we consider the quality of the tracking behavior via counting identity switches of tracklets, *IDS*, to asses tracking stability. Moreover, we report the percentage of active and lost tracklets for more than 80% of their life span, i.e., *MT* and *ML*, respectively.

C. IMPLEMENTATION DETAILS

We pre-trained our LMOT model on CrowdHuman [29], CityPersons [30], and ETH [31] for 60 epochs. Then, we trained LMOT on MOT16/17/20 datasets [32], [33], each dataset in a separate experiment, using Adam optimizer for additional 30 epochs with a learning rate of 10^{-5}, where the learning rate decays to 10^{-6} for the last five epochs with a batch size of 6. The input image resolution is 544 $\times$ 960, and the output feature heads resolution is 136 $\times$ 240. Moreover, the focal loss hyper-parameters α and β are 2 and 4, respectively. Besides, the total loss hyper-parameters λ_H and λ_D are set to 1, and λ_{SO} is set to 0.1. For the online data association, the threshold score for high-confidence detections is 0.3. Furthermore, as for the Jonker-Volegenant algorithm, we used a matching threshold of 0.8 and 0.7 in the first and second stages, respectively. We keep the unmatched tracklets for 30 future frames to account for possible occlusions. We performed all our experiments using a mobile Nvidia Geforce RTX 2070 discrete GPU with 2304 CUDA cores and 8 GB of RAM with a frequency of 1.4 GHz.

D. BENCHMARK RESULTS

We evaluated our approach compared to the state-of-the-art approaches, i.e., FairMOT [7] and GSDT [6], as well as the other top-performing methods [4], [5], [20], [21] on the testing sequences of MOT16/17/20 datasets. However, concerning the validation results, we compared our model to only those methods that reported their validation results based on half-splitting the training sequences of MOT17 [4], [5], [7] to have a fair comparison with our approach.

As for the validation phase, our approach significantly improves running time while outperforming all the other methods in tracking accuracy, as demonstrated in Table 3. Furthermore, LMOT achieves a comparable IDF1 score to FairMOT [7] while outperforming the other approaches. Additionally, LMOT demonstrates a robust tracking behavior compared to all the other approaches, which can be shown via the improvement in *IDS*, *MT*, and *ML* scores.

Concerning the testing phase, as shown in Table 4, LMOT significantly outperforms the state-of-the-art approaches regarding the runtime per frame in the private mode, which corresponds to being ten times faster compared to the state-of-the-art considering the average runtime over all the datasets. Furthermore, concerning the tracking accuracy, LMOT achieved a very close performance for MOT16/17 datasets leading to an accuracy of just 2% less than the state-of-the-art approach for each of these datasets. Additionally, LMOT achieved an accuracy of 8% less than the state-of-the-art for the challenging MOT20 that includes extreme scenarios with massive crowds. Similarly, LMOT achieved very close IDF1 scores for MOT16/17 datasets to the state-of-the-art while achieving an acceptable score for MOT20. It achieves second-best IDF1 for MOT16/17 datasets. Overall, LMOT achieved a significant improvement in runtime while achieving a comparable accuracy that is 3.8% less than the state-of-the-art on average. Thus, LMOT, as opposed to the state-of-the-art approaches, can operate in real-time applications while maintaining an acceptable tracking accuracy.

Furthermore, LMOT demonstrates a stable tracking behavior since it outperforms the state-of-the-art approaches for the number of identity shifts, i.e., IDS, for MOT16/20 datasets. The tracking behavior's stability can also be shown from the tracklets that were active or lost tracklets most of their life span, i.e., *MT* and *ML*. LMOT achieved second place for the percentage of mostly active tracklets and the second minimum percentage of lost tracklets for MOT16/17. Furthermore, as for MOT16, LMOT achieved comparable percentages to the state-of-the-art approaches. However, the tracking stability of LMOT was affected by the severe occlusions caused by the massive crowds of MOT20.

All in all, LMOT is, on average, ten times faster than the state-of-the-art while sacrificing only 3.8% of accuracy. Accordingly, LMOT represents a real-time tracking module while maintaining a robust performance in challenging situations, such as those depicted in Figure 4, compared to the other state-of-the-art approaches that severely trade-off computational runtime for robustness. For example, GSDT [6] and FairMOT [7] suffer from scalability issues and run much slower in crowded frames. Moreover, Trackformer [25] is computationally expensive to run and hard to train and converge due to using a complex vision transformer. Additionally, we demonstrate further examples of our real-world experiments via a demo at our supplementary materials.[2]

In addition to conventional evaluation, we evaluated LMOT on MOT16/17/20 datasets [32], [33] in a public detection mode to assess its tracking capabilities apart from its ability to carry out detections. In such mode, LMOT only performs tracking relying on a set of given detections provided by the MOT16/17/20 datasets and then generates tracklets for such detections. In this regard, we compared LMOT to TrackFormer [25] and TMOH [35], which are the top-performing approaches that support public detection mode. The state-of-the-art approaches do not support public detection mode due to using re-identification embeddings in data association; thus, it is not possible to decouple detection and tracking in their case. As shown in Table 5, LMOT demonstrates an outstanding runtime performance achieving a significant improvement. Furthermore, as for tracking accuracy, LMOT is superior in MOT16/17 datasets with an average improvement of 8% in MOTA and 6% in IDF1 while achieving a comparable performance in MOT20, i.e., 3.8% and 4.9% less than TMOH [35] for MOTA and IDF1, respectively. Additionally, LMOT demonstrates a smooth and robust tracking behavior achieving the highest percentage of mostly active tracklets and the lowest percentage of mostly

[2]https://scholar.cu.edu.eg/abayoumi/LMOT

TABLE 3. Validation results on MOT17 validation sequences running on RTX 2070 Mobile GPU.

Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
LMOT (ours)	**72.2**	74.3	**48**	**15.3**	**278**	**35**
FairMOT [7]	71.1	**75.6**	44.1	15.5	327	110
TraDes [5]	68.2	71.7	41.9	19.5	285	125
CenterTrack [4]	66.1	64.2	41.3	21.2	528	80

TABLE 4. Testing results on MOT16/17/20 testing sequences in private mode running on RTX 2070 Mobile GPU.

Dataset	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
MOT16	LMOT (ours)	73.2	72.3	44	17	**669**	**35**
	FairMOT [7]	**74.9**	**72.8**	**44.7**	**15.9**	1074	110
	GSDT [6]	74.5	68.1	41.2	17.3	1229	507
	TraDeS [5]	70.1	64.7	37.3	20.0	1144	125
	CTracker [20]	67.6	57.2	32.9	23.1	1897	65
	JDE [22]	64.4	55.8	35.4	20.0	1544	370
	TubeTK [21]	64.0	59.4	33.5	19.4	1117	2500
	DeepSORT [14]	61.4	62.2	32.8	18.2	781	200
	SORT [13]	59.8	53.8	25.4	22.7	1423	155
MOT17	LMOT (ours)	72.0	70.3	45.4	17.3	3071	**35**
	TrackFormer [25]	**74.1**	68	**47.3**	**10.4**	**2829**	700
	FairMOT [7]	73.7	**72.3**	43.2	17.3	3303	110
	GSDT [6]	73.2	66.5	41.7	17.5	3891	507
	TraDeS [5]	69.1	63.9	36.4	21.5	3555	125
	CenterTrack [4]	67.3	59.9	34.9	24.8	2898	80
	CTracker [20]	66.6	57.4	32.2	24.2	5529	65
	TubeTK [21]	63.0	58.6	31.2	19.9	4137	2500
MOT20	LMOT (ours)	59.1	61.1	25.1	23.0	**1398**	**35**
	FairMOT [7]	61.8	67.3	**68.8**	**7.6**	5243	150
	GSDT [6]	**67.1**	**67.5**	53.1	13.2	3133	600
	MLT [34]	48.9	54.6	30.9	22.1	2187	621

TABLE 5. Testing results on MOT16/17/20 testing sequences under the "public detector" protocol on 2D object tracking.

Dataset	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
MOT16	LMOT (ours)	**72.4**	**71.4**	**40.4**	**19**	744	**35**
	TMOH [35]	63.2	63.5	27	31	**635**	1600
MOT17	LMOT (ours)	**70.4**	**68.7**	**48**	**12**	6138	**35**
	TrackFormer [25]	62.3	57.6	29.2	27.1	4018	700
	TMOH [35]	62.1	62.8	26.9	31.4	**1897**	1600
	CenterTrack [4]	61.5	59.6	26.4	31.9	2583	80
MOT20	LMOT (ours)	56.3	56.3	30.9	22.1	**1671**	**35**
	TMOH [35]	**60.1**	**61.2**	**46.7**	**17.8**	2342	1600

TABLE 6. MOT17 validation set results showing the effect of using pre-training datasets.

Pre-training Datasets	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓
None	61.2	63.4	35.7	22.7	2250
CrowdHuman [29]	67.2	69.6	39.2	21.8	751
CrowdHuman [29] + CityPersons [30]	70.9	72.6	45.4	19.2	600
CrowdHuman [29] + CityPersons [30] + ETH [31]	**72.2**	**74.3**	**48**	**15.3**	**278**

lost tracklets for MOT16/17 datasets while maintaining the smallest number of identity shifts in MOT20. Therefore, LMOT demonstrates a high tracking performance apart from its ability to carry out detection.

E. ABLATION STUDIES

Furthermore, we investigated the effect of our design choices. First, we validated the significance of performing pre-training using CrowdHuman [29], CityPersons [30] and ETH [31]

TABLE 7. MOT17 validation set results comparing using different backbone encoder networks in LMOT for generating detection feature maps.

Encoder	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
Simplified DLA-34 (ours)	72.2	74.3	48	15.3	**278**	35
DLA-34	**72.3**	**74.4**	**49.6**	**14**	400	55
Resnet-34	63.8	65.6	35.2	25.4	300	**18**
Resnet-50	64.5	68.4	36.8	22.3	441	72

TABLE 8. Impact of using prior heatmaps as an input to the linear transformer module on MOT17 validation set.

Model	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
LMOT (without heatmaps)	69.1	71.8	43.2	20.7	600	**27**
LMOT (with heatmaps)	**72.2**	**74.3**	**48**	**15.3**	**278**	35

TABLE 9. Impact of using different matching algorithms for online data association on MOT17 validation set.

Algorithm	MOTA↑	IDF1↑	MT↑	ML↓	IDS↓	Time (ms)↓
Jonker-Volegnant [28]	**72.2**	**74.3**	**48**	15.3	**278**	35
Greedy [4]	70.4	68.4	47.2	**14.2**	1881	**33**
Hungarian [36]	70.8	64.2	46	14.5	1569	37

datasets on the MOT17 half-split validation sequences. Thus, we considered different pre-training scenarios for LMOT. As shown in Table 6, increasing the number of pre-training datasets of limited crowds improves both the tracking accuracy and behavior. Such improvement is due to the boost in the detection capabilities of LMOT caused by warming up our model and allowing it to learn necessary features in limited crowds before exposing it to complex scenarios.

Additionally, we explored the significance of our proposed simplified DLA-34 network by replacing it in LMOT with different encoder networks. We evaluated that on the validation sequences of MOT17, as shown in Table 7. Accordingly, we compared our simplified DLA-34 to the original DLA-34 and the commonly used Resnet-34 and Resnet-50 networks [23]. DLA-34 achieves the best MOTA and IDF1 scores; however, our simplified DLA-34 achieves slightly less MOTA and IDF1 scores while significantly improving runtime per frame, i.e., approximately 1.5 times faster. On the other hand, Resnet-34 and Resnet-50 fail to achieve a comparable performance, which consumes even more time in Resnet-50. Therefore, our proposed simplified DLA-34 improves the runtime while maintaining close detection and tracking performance.

Moreover, we show the impact of using prior heatmaps as an input to the extended linear transformer. As shown in Table 8, prior heatmaps improve the tracking accuracy and behavior while slightly trading-off the runtime. In other words, prior heatmaps help generate crucial tracking features that boost the accuracy while not affecting the runtime performance of the model.

Finally, we evaluated the effect of using Jonker-Volegnant algorithm [28] for our proposed online data association technique compared to the commonly used Hungarian [36], and greedy [4] techniques. As shown in Table 9, Jonker-Volegnant algorithm [28] surpasses all other matching algorithms with a 2% and 6% increase in MOTA and IDF1, respectively, due to its ability to reduce the number of false positives. Moreover, Jonker-Volegnant achieves the lowest IDS compared to other matching algorithms. Furthermore, as for the runtime, the performance of Jonker-Volegnant algorithm is average compared to the other approaches.

V. CONCLUSION

This paper introduces a novel real-time multi-object tracking approach entitled LMOT. We evaluated our approach on the well-known challenging MOT16/17/20 benchmarks, demonstrating a robust tracking performance that outperforms the state-of-the-art approaches from a runtime perspective while maintaining very close and comparable accuracy. Furthermore, our experiments showed the robustness of LMOT to occlusions and its ability to recover from them, relying on our two-stage online data association technique. Furthermore, our simplified DLA-34 network generates powerful detection features while boosting the runtime. Moreover, our extended heatmaps-based linear transformer supported the detection features with robust tracking features in linear complexity. However, as for future work, we will extend our model to be able to deal with illumination and camera viewpoint challenges, which severely affect tracking robustness in massive crowds.

REFERENCES

[1] C. Luo, C. Ma, C. Wang, and Y. Wang, "Learning discriminative activated simplices for action recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 4211–4217.
[2] M. Yang, Y. Wu, and Y. Jia, "A hybrid data association framework for robust online multi-object tracking," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5667–5679, Dec. 2017.
[3] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
[4] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2020, pp. 474–490.
[5] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12347–12356.

[6] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13708–13715.
[7] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 3069–3087, Sep. 2021.
[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
[9] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross, "Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking," *Robot. Auto. Syst.*, vol. 54, no. 9, pp. 721–728, Sep. 2006.
[10] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using K-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
[11] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
[12] M. Munaro, F. Basso, and E. Menegatti, "OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks," *Robot. Auto. Syst.*, vol. 75, pp. 525–538, Jan. 2016.
[13] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
[14] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
[15] S.-H. Bae, "Online multi-object tracking with visual and radar features," *IEEE Access*, vol. 8, pp. 90324–90339, 2020.
[16] T. Liang, L. Lan, X. Zhang, X. Peng, and Z. Luo, "Enhancing the association in multi-object tracking via neighbor graph," *Int. J. Intell. Syst.*, vol. 36, no. 11, pp. 6713–6730, Nov. 2021.
[17] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[19] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.
[20] J. Peng, C. Wang, F. Wan, Y. W. Yang Wu, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 145–161.
[21] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6307–6317.
[22] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 107–122.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[24] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[25] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8844–8854.
[26] L. Melas-Kyriazi, "Do you even need attention? A stack of feed-forward layers does surprisingly well on ImageNet," 2021, *arXiv:2105.02723*.
[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[28] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, Nov. 1987.
[29] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.
[30] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3213–3221.
[31] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[32] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.
[33] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.
[34] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong, "Multiplex labeling graph for near-online tracking in crowded scenes," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7892–7902, Sep. 2020.
[35] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10953–10962.
[36] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.

RANA MOSTAFA was born in Cairo, Egypt, in 1996. She received the bachelor's degree in computer engineering from Cairo University, Egypt, in 2019, where she is currently pursuing the master's degree. She worked as a Software Application Engineer at Valeo. She is particularly interested in deep learning with robotics and development of autonomous systems.

HODA BARAKA is currently a Professor of computer engineering at the Faculty of Engineering, Cairo University, as well as an Advisor to the Egypt Minister of ICT for Technology Talents Development. She has been awarded the UNESCO King Hamad Bin Isa Al Khalifa prize for the use of ICTs in education, for her contribution in the development and enhancement of education in Egypt using ICT. She has more than 30 years of extensive experience as a consultant in the field of digital transformation for public and private sectors. She published more than 50 articles in the fields of computer networks and natural language processing. She was a member of WEF/UNESCO Partnership for Education Technical Advisory Group on Capacity Development and a member of the WEF Global Agenda Council for Technologies in Education.

ABDELMONIEM BAYOUMI (Member, IEEE) received the Ph.D. degree from the University of Bonn, Germany, in 2018. He is currently an Assistant Professor with the Department of Computer Engineering, Cairo University. He works as the Head for the Robotics Track at Digital Egypt Builders Initiative (DEBI) in cooperation with the University of Ottawa, Canada, and the Egyptian Ministry of Communications and Information Technology. Previously, he worked at the Humanoid Robots Laboratory, University of Bonn. He published several papers in top robotics conferences and journals. His research interests include cognitive robotics focusing on navigation problems and deep learning.