https://doi.org/10.2352/EI.2022.34.6.IRIACV-274

Quantitative analysis of Deep learning Based Multi-Target Tracking Algorithms

Sanam Nisar Mangi, Mohib Ullah, Faouzi Alaya Cheikh
Department of Computer Science
Norwegian University of Science and Technology

***Abstract*—Multi-object tracking is an active computer vision problem that has gained consistent interest due to its wide range of applications in many areas like surveillance, autonomous driving, entertainment, and, gaming to name a few. In the age of deep learning, many computer vision tasks have benefited from the convolutions neural network. They have been optimized with rapid development, whereas multi-target tracking remains challenging. A variety of models have benefited from the representational power of deep learning to tackle this issue. This paper inspects three CNN-based models that have achieved state-of-the-art performance in addressing this problem. All three models follow a different paradigm and provide a key inside of the development of the field. We examined the models and conducted experiments on the three models using the benchmark dataset. The quantitative results from the state-of-the-art models are listed in the standard metrics and provide the basis for future research in the field.**

***Index Terms*—Multi target tracking, Deep learning, Computer vision**

I. Introduction

Multi-target tracking has captivated researchers from last decades due to its applications in various disciplines, including human-computer interaction [2], [7], video surveillance [14], [23] and virtual reality [17], [19]. Multi-target tracking has a variety of applications in the areas like pose estimation [8], [13], behavior analysis [6], [18], surveillance [10], [20] and security. Despite the numerous ways [3], [21], [22] that have been presented to address this problem, it remains a difficult challenge to solve. In general, MTT algorithms assign each identified object a unique id that remains unique to the object for a certain period. Motion trajectories for objects to be tracked are constructed using these Ids. The efficiency of target tracking is highly dependent on the precision of the object detection system. The challenge of MTT could be the product of different occlusions and interactions between objects, which may also be identical in appearances in addition to problems of background clutter, pose changing, initialization, and termination of tracks. Many algorithms are proposed to achieve robust tracking. There are a variety of datasets available to test the algorithms and draw comparisons between them. Since deep neural networks (DNNs) can retrieve abstract and complex features by learning rich representations from inputs, they have been utilized in top-performing MOT algorithms, aiding in resolving the subtask in which the problem is divided. There is a broad scope of approaches introduced in the field of MTT. However, most MTT algorithms follow all/a piece of the steps identified by [27] as follows:

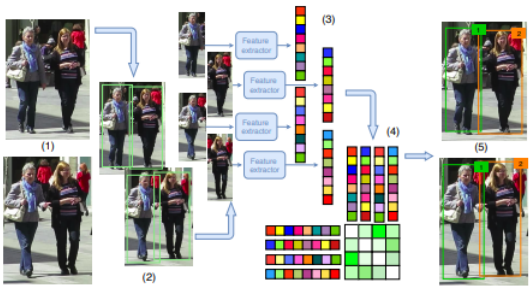

Fig. 1. Illustrates the usual work flow of MOT algorithm involving detection, motion prediction, affinity stage and association step. [27]

- The task involves identifying and locating objects (belonging to the target class) in each input frame by creating bounding boxes around those objects or labeling each pixel containing that object, called detection.
- Motion prediction stage: The algorithm extract motion features by analyzing the detections or tracklets. Additionally, the motion predictor could determine the following location of each object being tracked in the same step.
- Affinity Stage: Predictions computed in the previous stage are used to determine the distance score or similarity between tracklets/detection pairs.
- Association Stage: This step involves linking detections with tracklets that belong to the same target and assigning the ID to those that identify the same target by utilizing the distance score/ similarity.

After inspecting recent developments in the field of multi-target tracking, we experiment on three multi-target tracking algorithms that make use of deep learning capabilities on the MOT17 dataset and provide a performance comparison between them. This paper will briefly describe the implementation of the MTT algorithms, details of the conducted experiments, the data set used to conduct the experiments, and the metrics used to evaluate results. We then summarize experimental results to provide quantitative comparisons and to point to some critical observations by evaluating these results.

II. Methods

The table 1 illustrates the information of Methods that's are investigated end evaluated in this paper.

TABLE I
STATE-OF-THE-ART CNN BASED MOT ALGORITHMS

Year	Full Name	Mode	Weblink
2019	Deep Affinity Network for Multiple object tracking	Online	https://github.com/shijieS/SST.git.
2019	Towards Real-Time Multi-Object Tracking	Online	https://github.com/Zhongdao/Towards-Realtime-MOT
2019	Tracking without bells and whistles	Online	https://git.io/fjQr8

A. *Deep Affinity Network (DAN) for MOT*

ShiJie at el. [16] proposed an online tracking algorithm called Deep affinity based network (DAN) that models target appearances in association with their affinities across two different adjacent or non-adjacent video frames. The overall method takes advantage of CNN's effective affinity estimation in order to link targets in the current frame to those in several previous frames to measure accurate trajectories. The figure 2 illustrates the DAN architecture, which is divided into two components a) Feature extractor and b) Affinity estimator. The entire network in trainable from end to end. For the training it requires, pair of video frames and its object centers. The network does not restrict two frames to appear consecutively in video instead it allow frames to be n timestamps apart. Whereas DAN network is ultimately deployed to track objects in consecutive video frames, training them with non-consecutive frames benefits the overall approach in accurately associating objects in a given frame to those in multiple previous frames.

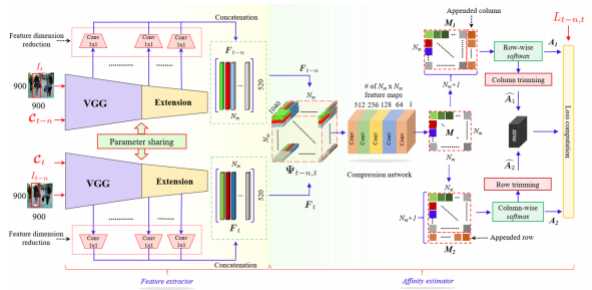

Fig. 2. Semantics illustration of Deep Affinity Network (DAN) [16]

The main components are briefly discussed as follow:

- **Feature Extractor** performs an extraction process by passing two video frames and object centers through two streams of convolution layers. These streams are based on VGG architecture, having fully linked and softmax layers converting to convolution layers. As per the available datasets, the size of the input frame is also kept wide as compare to original VGG16. The 36-layerd extension network is preferred as reduces feature map size to 3 x 3 and results in better performance, also comprehensive appearance modeling is ensured at multiple abstraction layers. As it has wide receptive field, the latter layers of extension sub network provide better modeling of the object surrounding, which increases overall performance. Moreover, knowing the object center location allows the model to extract center pixels of the object as their representative features.
- **Affinity estimator** The key role of this component of DAN is the computation of affinities among objects by utilizing extracted features. It consists of a compression network, having 5 convolution layers with 1 x1 kernels. The network projects a tensor that computes object features combination to encode similarities between the features between objects, which contributes to the adjacent objects not to get influenced by the feature maps.

The DAN network operates in such a way that it requires a single frame as its input, along with the location of the object center. The feature matrix is computed by the feature extractor for the input frame and proceeds to the affinity estimator, which uses the feature metrix to determine the tensor for the frame pair. Thus each frame is processed through an object detector and a feature extractor only once but the features are used numerous times for computing affinities with several other frames in pairs. The comprehensive appearance modeling and effective affinity computation is the strength of DAN approach. This model claims to be the first deep network models that have computed object appearance and computed inter frame object affinities simultaneously which has sets its performance apart from other methods.

B. *Towards Real-Time Multi-Object Tracking*

Since multi-target tracking usually divided into the step of localizing object in frames and assigning trajectories to those objects, which requires system to have two components i.e. detector and embedding model (termed as Separate detection and Embedding – SDE methods), the overall inference time which is summation of the two components increases as target increase. Such models bring critical efficiency issues when designing Real time MOT systems. Wang et al. [24] put forward Joined detection and embedding model (JDE) that enables the detection and targets to be learned in a shared model thus re-computation is avoided. It works in a real time and as precise as other SDE methods.

1) Architecture Overview: The JDE architecture is based on the Feature Pyramid Network (FPN) which gives better results in situations where the target scale varies. FPN predicts target from multiple scales. JDE networks obtain feature map of video frame using backbone network at 3 different scales, the feature maps are up scaled and fused, then the predictions heads are added upon these fused features at all three scales. Prediction head a stacked convolutional layers and produces a dense prediction map. The detection branch of JDE is implemented in the same way as the RPN [11], with a few improvements in the configuration of the anchors in terms of numbers, sizes and aspect ratio, and the selection of the required threshold value for the foreground/background

allocation. Thresholds are useful and efficiently eliminate false alarms, which typically occur under heavy occlusions. The two loss functions that are part of the learning objective of detection are foreground/background classification loss and bounding loss of regression, and described as a cross-entropic loss and a smooth loss respectively. Regression targets are encoded in the same way as the RPN.

To achieve an embedded learning, the model used triple loss. However, formulation of triple loss has problem of huge sampling space in the training set. Therefore, the model look at the mini batch and mine all the negative samples and the hardest positives sample in this mini- batch. Figure 3 shows architecture of JDE model.

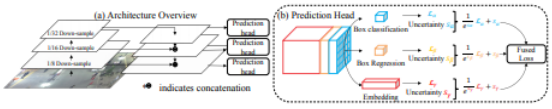

Fig. 3. Illustrates Network architecture (a) and the prediction head (b) for JDE model [24]

C. Tracking without bells and whistles

The tracking problem consist on several challenging tasks that include object re-identification, motion prediction and dealing with occlusions. Tracking by detection paradigm is considered as the most preferred paradigm to solve the problem of tracking objects in multiple frames, and is divided into the detection step and linking step. The linking step i.e. forming association with corresponding detections across time is challenging on its own due to lost and false detections, occlusions and interactions between the targets in crowded environments. To address these challenges, researchers have developed increasingly complicated models with only little improved performance. Philipp at el. [1] proposed a method for tracking without focusing on any of these specific tasks, i.e., the training data was not trained or optimized, and the tracking results were obtained by training a neural network solely on the detection task. The model uses the bounding box regression of an object detector to forecast the position of an item in matching frames, effectively converting the detector into a tracktor.

Figure 4 illustrates the working of MTT only with the object detector in two steps. The object detector regression adjusts the old frame t-1 track bounding boxes to the new location of the item at frame t in the first step. The modified bounding box positions' matching object classification scores are then utilized to remove possibly occluded tracks. This strategy has two major advantages. This tracker is online since it does not require tracking-specific training and does not do extensive optimization during the test period.

1) Architecture overview: The regression based detector is the core element of the tracking pipeline. For detection task, Faster R-CNN is utilized to apply a Region proposal network that generate a multitude of bounding boxes for each targeted

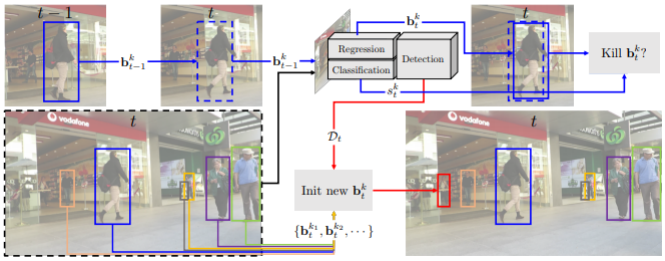

Fig. 4. Illustration of tracktor achitecture

object. For each proposal, feature maps are extracted using Region of interest (ROI) pooling [5], followed by passing through the classification and regression heads. The proposal was then given an object score by a classification head, a bounding box refined by a regression head in close proximity to an object. The detector then applies non-maximum suppression (NMS) to refined boundary box proposals, and then returns the final set of object detection.

Extracting trajectories of objects in a video sequence, is a challenge of multi-object tracking. The model sub-divide the task into the steps of bounding box regression and track initialization. During the first step, the bounding box regression is utilized to extend active trajectories to current frame i.e. applying ROI pooling on current frame features with coordinates of previous bounding box. The identity is transferred from the previous one to the regressed bounding box automatically, forming trajectory efficiently, this apply for all the subsequent frames. The second step involves bounding box initialization. The model consider new trajectory for detection only if it is covering new object, which does not belong to any previous trajectory.

It is worth noting that the model relies entirely on the object detection process, and no further training or optimization of complex tracking is needed. This enables tracker to directly benefit from enhanced object detection methods and most significantly, allow a relatively cheap transition to various tracking datasets or situations where there is no ground truth but just detection data available.

III. Experiment

This section will evaluate the above discussed models on well-known object tracking challenge i.e. MOT17 dataset. The evaluation of methods were performed on IDUN computing cluster [15] . The cluster has over 70 nodes and 90 GPGPUs. Each node contains two cores of Intel Xeon, at least 128 GB of main memory, and is connected to the Infiniband network. Half the nodes are fitted with two or more Nvidia Tesla P100 or V100 GPGPUs. Idun's storage is supported by two storage arrays and a parallel distributed Lustre file system.

A. Implementation details

DAN was implemented using Pytorch framework [26], Nvidia Tesla P100 GPU is used to perform training. In order to

optimize the hyper-parameters of DAN, MOT17 [12] dataset , due to its manageable size, is utilized. Validation set to train the model was specified. The hyper-parameters values are set such that the Batch size = 8, number of epochs per model training = 90, the maximum number of objects allowed per frame = 80. All the data for both train and test were resized to 900 X 900 as per the input frame size. Learning rate for a network is set to 0.01.

For JDE, DarkNet-53 [9] was utilized as backbone network. The network is trained with epochs = 90, learning rate is set to 0.01. To minimize over-fitting, numerous data improvement methods, for instance random rotation, random scale and color jittering, are used to minimize over-fitting. The augmented images are eventually adjusted to a defined resolution. The input resolution is set as 1088 × 608.

For tracktor, since this model perform tracking with object detection method with no dedicated training or optimization on ground truth data, we use pre-trained (FRCNN) [52] multi-object detector with Feature Pyramid Networks (FPN) as it provided with [1] implementation, batch size = 1, number of extracted regions are R = 256

B. Multiple Object Tracking 17 (MOT17)

Some benchmarks are established to facilitate research areas in order to obtain an objective measure of performance. One of the most recent benchmarks is the MOT17 (Multiple Object Tracking 17) challenge. It consists of a series of 14 video sequences of various indoor and outdoor crowded scenarios, from different points of view, different camera motion and under different weather conditions. Every sequence is provided with three sets of detections: DPM [25], Faster-RCNN [11], and SDP [26]. Each sequence is divided into two clips, one for each training and testing. Both online and offline tracking approaches are accepted by this challenge, where the frames are allowed to use future video frames in order to make predictions of tracks.

C. Dataset

The MOT17 dataset comprises of seven training videos provided with their ground truth tracks and detections from three detectors. The scenes differ considerably in terms of lighting conditions, background and view point of the camera. They differ from one another in terms of frame rate, number of objects per frame, and number of tracks, which make the dataset difficult. The key attributes of the MOT17 training dataset are summarized in Table II.

D. MOT Evaluation metrics

In order to evaluate performance of MOT approaches, evaluation metrics are required as they provide quantitative comparison. Evaluation of MOT models are not straight forward and different components and parameters impact on overall performance, hence it is essential to measure the impact.

The evaluation metrics that we used to compute the results and evaluate the performance are listed in table iii.

Metrics for tracking are categorized into subcategories by different attributes and defined by [32] as follows:

TABLE II
MOT17 TRAINING DATASET ATTRIBUTES [12] DETECTED BOXES GIVEN THE DETECTORS ARE LISTED IN LAST THREE COLUMNS, 'DENSITY' REPRESENTS THE AVERAGE NUMBER OF OBJECTS PER FRAME, WHILE 'MOVE' SPECIFIES IF THE MOVING CAMERA WAS USED WHILE RECORDING THE VIDEO. [16]

Video Index	Resolution	FPS	Tracks	Density	Move	DPM [4]	SPD [26]	FRCNN [11]
02	1920X1080	30	62	31.0	N	7267	11639	8186
04	1920X1080	30	1050	45.3	N	39439	37150	28406
05	640X480	14	133	8.3	Y	4333	4767	3848
09	1920X1080	30	26	10.1	N	5976	3607	3049
10	1920X1080	30	57	19.6	Y	8832	9701	10371
11	1920X1080	30	75	15.5	Y	8590	7509	6007
13	1920X1080	25	110	8.3	Y	5355	7744	8442

TABLE III
MOT METRICS USED FOR BENCHMARKING, THE UP ARROW (RESP. DOWN ARROW) INDICATES THE BETTER PERFORMANCE WHEN QUANTITY IS GREATER (SMALLER RESPECTIVELY)

Metric	Description	
Precision	Ratio of precise detection to the total detections	↑
FAF	False alarms per frame in a sequence	↓
MODA	Combined missed detection with false alarm ratio	↑
MODP	Average overlap between true positive and ground truth	↑
MOTA	Overall tracking accuracy	↑
IDS	Id switches	↓
MOTP	overlap between the estimated positions and ground truth	↑
MT	Mostly tracked targets	↑
ML	Mostly lost targets	↓
Rcll	Mostly lost targets	↓
IDF1	The percentage of detected targets	↑
FP	Number of false positives	↓
FN	Number of false negatives	↓

- **Accuracy:** These metrics measure the accuracy of an algorithm in terms of tracking object. The ID switches (IDs) metric counts the number of times an algorithm switches between object. The Multiple Object Tracking Accuracy (MOTA) metric calculates the overall tracking performance by combining the rate of false positive, false negative and mismatch into a single quantity. MOTA is widely accepted evaluation metrics even though there are some drawbacks.
- **Precision:** The precision of the tracked objects are measured by bounding box overlap and distance, which is measured by Multiple Object Tracking Precision (MOTP), OSPA and Tracking Distance Error (TDE). Completeness: The three metrics Mostly tracked (MT), Mostly Lost (ML) and Partly Tracked (PT) are completeness metrics and indicate if the ground truth trajectories are tracked completely.
- **Robustness:** The ability of recovering from occlusion of MOT algorithm is evaluated by metrics known as Shortterm occlusion (RS) and Recover from Long-term

occlusion (RL).

IV. Results

We Train DAN network with training data set provided by MOT 17 dataset, and perform the test on 12 different train sequences. The results were obtained on MOTA, FN,FP, ID_SW evaluation metrics and summarizes in table IV.

TABLE IV
MOT Metrics used for bench-marking, The up arrow (resp. down arrow) indicates the better performance when quantity is greater (smaller respectively)

Sequence	MOTA ↑	FP ↓	FN ↓
MOT17-11-DPM	43.07%	794	4492
MOT17-11-FRCNN	56.44%	303	3740
MOT17-10-SDP	59.12%	1229	3724
MOT17-04-SDP	73.87%	919	11362
MOT17-05-SDP	53.78%	413	2567
MOT17-10-FRCNN	41.55%	2012	5169
MOT17-04-FRCNN	51.56%	1891	21068
MOT17-13-SDP	31.45%	1837	5842
MOT17-04-DPM	32.21%	3586	28322
MOT17-05-FRCNN	47.71%	222	3294
MOT17-11-SDP	66.19%	561	2511
MOT17-09-FRCNN	55.44%	30	2310

In particular, DAN shows significantly better results on widely accepted MOT metrics i.e. when evaluating on MOTA and MOTAL. The overall DAN results on MOT17 test tests are summarized in table V.

TABLE V
Evaluation of DAN on MOT17 test-set

	MOTA ↑	MOTAL ↑	MOTP ↑	Rcll ↑	IDF1 ↑
DAN	52.4224	53.916	76.9071	58.4225	49.4934

It's in Fig. 5, we're showing two examples of DAN Performance tracking on the MOT17 challenge. The results are from the tracking test, the colorful bounding boxes in the frames are shown, showing the trajectory,predicted by DAN.

Fig. 5. Visual tracking results of DAN on MOT17 video sequence

Since tracktor [1] does not require any tracking specific training, We used Faster R-CNN set of MOT17 public detections to evaluate this model.We train DAN [16] and JDE [24] models with MOT17 datasets and obtain the results on multple metrices. However, for comaprison, we are considering 7 metrics, and the results accumulated over all sequences are summarized in the table VI

Fig. 6. Visual tracking results of DAN on MOT17 video sequence

TABLE VI
Evaluation of Tracktor [1], DAN [16], JDE [24] on MOT17 benchmark, The symbol ↑ indicates better performance are when values are higher, and ↓ implies the performance improvement when values are lower. Lower values are favored.

	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↑	IDsw
Tracktor	53.5%	52.3%	19.5	36.6	12201	248047	2072
DAN	52.45%	49.49%	21.4	30.7	25423	234512	8431
JDE	63.1%	68.4%	28.4	33	4886	21504	1258

The JDE network achieves the highest MOTA score of 63.1%, however it also outperform IDF1, MT and F evaluation metrics. DAN achieves best performance on ML and FN metrics.

Considering the overall tracking accuracy, JDE is performance efficient as compare to the other two methods that we considered in the study.

V. Result Discussion

The Deep Affinity Network (DAN) works on a paradigm of tracking by detection, it learns comprehensive features of pre-detected objects at different level of abstractions, and for those features, it perform pairing permutations in two frames to form object affinities. It is the first deep network to model the appearance of the object and the computed affinities for the inter-frame object. This property set DAN apart from other models, which enables DAN tracker to attain high overall performance with the high Multiple Object Tracking Accuracy on all challenges, it achieves 53.5% MOTA during this evaluation. During our evaluation. There are some situations e.g. highly crowded scenes where DAN compromise its performance when the frames feature identical objects in a scene at a very close position at multiple time stamps. This often resulted in ID switches between objects.

The Tracktor, on the other hand, which works on regression based detector, has demonstrated the ability to cope with detection by achieving 53.5% MOTA score, 52.3% for IDF1. However, without any prior training, this model is not expected to excel in crowded and occluded scenarios. This method of less-complex scenarios can be a motive for researchers to consider this model in order to refine it to work in more complex scenarios.

JDE, which outperform the other two models, and work on a paradigm of joined detection and embedding model. Considering the MOTA metric, the JDE provides competitive

tracking accuracy with MOTA score 63.1%. Other than that, the IDF-1 score, which reflects the performance in terms of association, JDE is also competitive with strong dataset combination. JDE can lead to show lots of ID switches, in case of inaccurate detection when multiple targets have large overlaps with each other, which compromises the IDF1 score. This is considered as a future work to be solved how to improve JDE to make more accurate predictions when there is a significant amount of target overlapping.

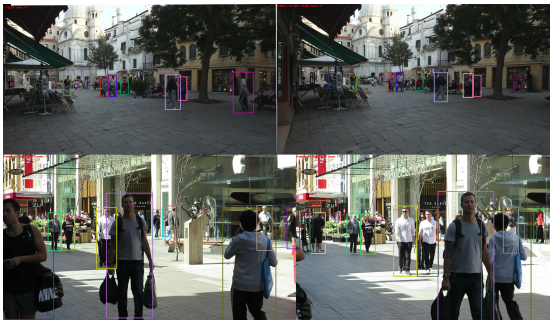

Fig. 7. Visual tracking results of JDE on two different MOT17 video sequence

VI. Conclusion

We presented a brief overview of multi-target tracking framework architectures and their critical components. Specifically, we focused on three novel methods and evaluated their performance on a standard benchmark dataset. All three methods are based on CNN but follow a different paradigm. These approaches have been reported to achieve state-of-the-art performance compared with similar implemented approaches. Out of three models, the JDE model achieved overall performance with the highest multiple objects tracking accuracies on the MOT17 challenge. We reported the MOTA, IDF1, MT, ML, FP, FN metrics. The paper provided an intuitive guide and an overview of the field.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE international conference on computer vision*, pages 941–951, 2019.

[2] Joshua Candamo, Matthew Shreve, Dmitry B Goldgof, Deborah B Sapper, and Rangachar Kasturi. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, 11(1):206–224, 2009.

[3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021.

[4] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.

[7] Junhao Huang, Zhicheng Zhang, Guoping Xie, and Hui He. Real-time precise human-computer interaction system based on gaze estimation and tracking. *Wireless Communications and Mobile Computing*, 2021, 2021.

[8] Akif Quddus Khan, Salman Khan, Mohib Ullah, and Faouzi Alaya Cheikh. A bottom-up approach for pig skeleton extraction using rgb data. In *International Conference on Image and Signal Processing*, pages 54–61. Springer, 2020.

[9] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30:386–396, 2017.

[10] Kleberson Meireles de Lima and Ramon Romankevicius Costa. Cooperative-phd tracking based on distributed sensors for naval surveillance area. *Sensors*, 22(3):729, 2022.

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[12] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[13] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[14] Hamza Riaz, Muhammad Uzair, Habib Ullah, and Mohib Ullah. Anomalous human action detection using a cascade of deep learning models. In *2021 9th European Workshop on Visual Information Processing (EUVIP)*, pages 1–5. IEEE, 2021.

[15] Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. Epic: An energy-efficient, high-performance gpgpu computing research infrastructure. *arXiv preprint arXiv:1912.05848*, 2019.

[16] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[17] Hideaki Uchiyama and Eric Marchand. Object detection and pose tracking for augmented reality: Recent approaches. 2012.

[18] Habib Ullah, Ihtesham Ul Islam, Mohib Ullah, Muhammad Afaq, Sultan Daud Khan, and Javed Iqbal. Multi-feature-based crowd video modeling for visual event detection. *Multimedia Systems*, 27(4):589–597, 2021.

[19] Habib Ullah, Sultan Daud Khan, Mohib Ullah, and Faouzi Alaya Cheikh. Social modeling meets virtual reality: An immersive implication. In *International Conference on Pattern Recognition*, pages 131–140. Springer, 2021.

[20] Mohib Ullah and Faouzi Alaya Cheikh. A directed sparse graphical model for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1816–1823, 2018.

[21] Mohib Ullah and Faouzi Alaya Cheikh. Deep feature based end-to-end transportation network for multi-target tracking. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3738–3742. IEEE, 2018.

[22] Mohib Ullah, Maqsood Mahmud, Habib Ullah, Kashif Ahmad, Ali Shariq Imran, and Faouzi Alaya Cheikh. Head based tracking. *Electronic Imaging*, 2020(6):74–1, 2020.

[23] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1):3–19, 2013.

[24] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019.

[25] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z Li. The fastest deformable part model for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2504, 2014.

[26] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[27] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017.