Chapter 4

An end-to-end Hybrid method for Multi-Object Tracking

This chapter describes the deep learning-based model developed for multi-object tracking. The model is introduced in Section 4.1. The description of the proposed model is given in Section 4.2. Section 4.3 gives the result generated by the proposed model and its analysis. Section 4.4 concludes the chapter.

4.1 Introduction

Various computer vision applications such as the Robotics field, self-driving cars and video surveillance needed multi-object tracking (MOT) [197] [198] to accomplish their tasks, which is a very crucial perception technique i.e., MOT. Tracking-by-detection is a useful technique that provides the detection results in each video frame and the association between the same targets of two frames. Tracking-by-detection is still a computer vision challenging task due to occlusion, irregular pattern of motion and training data lacking. Tracking-by-detection is still a challenging issue for computer vision applications because of occlusion and irregular motion patterns.

Deep-learning-based MOT approaches used models separately to calculate motion features and appearance features for target association. Deep Neural Networks have the powerful capability of representation to improve the MOT performance significantly by using target-wise motion features and spatial features for target association. Although, the computation task for the pair of features is hard to efficiently scale up for an indefinite number of objects in video frames. For instance, comparing appearance features among the variable number of targets between the frames of the video, there is one need for an indefinite number of iterative processes using fixed structured DNNs.

Practically, various state-of-the-art solved the MOT problem in two separate steps: 1- using two different models for motion and appearance features from input frames. 2- Calculated target association between the video frames from motion and appearance features. The Hungarian algorithm [199] is used to find one-to-one matches between the targets of a different frame. However, calculating high discriminative features usually needs a heavy DNN to be one step optimized for an end-to-end framework. Without a high-quality feature map, various other processes are needed to perform target association steps. Therefore, developing an efficient MOT system is very difficult when the two different models are used for feature extraction and data association.

In this thesis, the implemented model is a single-shot deep-learning-based approach for the above issues with an efficient technique. The implemented model uses an optical flow to extract motion features for targets from pixel level and spatial features that calculate the relative scale between the detected and tracked object for the target association. Then refinement and fusion of targets are applied to these features rather than using two inaccurate sources target proposals. The training process of the proposed model calculates the regression loss and classification loss to update the weight for the proposed network. As a result, the optical flow method extracts the motion-wise target features and the post-processing technique is used

to find a more accurate target tracking result. Given the 2 frame sequence as input, the proposed model generates association results for the tracking process directly.

4.2 Proposed Method and Model

4.2.1 A Hybrid method for Multi-Object Tracking

An end-to-end model that directly provides the result for both target motions and association for MOT. The Implemented method is a hybrid technique that uses flowNet-2 deep-learning model that calculates the target-wise motions for an indefinite number of objects from pixel-level optical flows. A matching technique directly associates the objects of two consecutive frames detection and tracking. The proposed model achieves state-of-the-art results over the existing methods.

4.2.1.1 Motion Estimation

To implement an end-to-end model, necessity is combined with the process of motion estimation and object association for a large number of objects present in video frames. The proposed model used two techniques combined to fulfill of above requirements. This technique estimates the motion at every pixel of each object presented in the video frame instead of individual feature extraction of objects. The term dense optical flow is used for the FlowNet-2, the combination of FlowNet-S (simple) and FlowNet-C (corner). The FlowNet-2 is responsible for the large quality improvement and speed too. This optical flow technique is useful to share the appearance matching at pixel-level overall objects.

The optical flow is used to estimate the motion of the presented object in the video frame at the pixel level. The architecture of the network is like flownet that has several convolutional layers and a fully connected layer for motion estimation from the previous frame to the current frame. The trained network takes two frames as input and produces outputs for many object motions from two consecutive input frames. The previous frame represented as $F_{t-1} \epsilon I^{3 \times b \times h}$

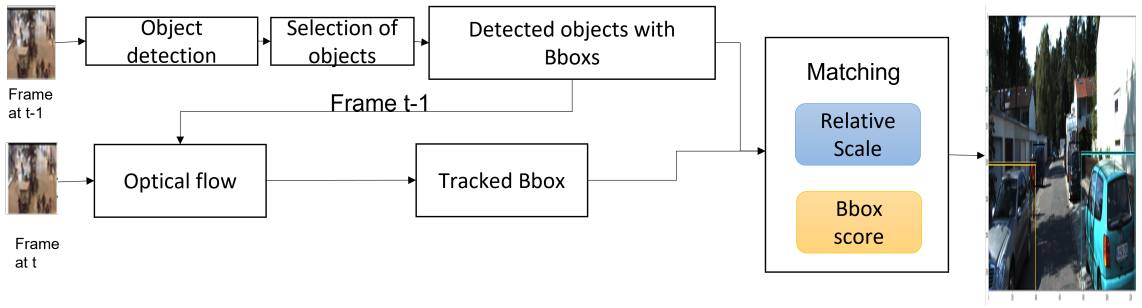

FIGURE 4.1: The framework of the proposed model for multi-object tracking that defines the two parallel process of detected and tracked object and match these two of two consecutive frame t-1 and t, with relative scale

and the current frame is represented as$f_t \epsilon I^{3\times b\times h}$ where $t > 1$ represented time index. First, The proposed model takes two frames as input for flownet-2 network and a pixel-level optical flow $O_t \epsilon F^{2\times b\times h}$ can be achieved at the output layer. Later the output of optical flow is used as an input and train it with the motion vectors of ground truth. Practically, the precedent frame contains multiple objects to be tracked in the current frame. For clarity, The objects of previous frame are denoted as $O_{t-1} = ok1_{t-1}, ok2_{t-1}, \ldots..okn_{t-1}|1k1, k2, \ldots..knN$ where $ok_{t-1} = (xk_{t-1}, yk_{t-1}, bk_{t-1}, hk_{t-1})$ denotes the position of $k \epsilon k1, k2 \ldots\ldots kn$ object in frame F_{t-1}. The corresponding objects in the present frame are represented by $O_t = ok1_t, ok2_t, \ldots\ldots..okn_t|1k1, k2, \ldots..knN$, where $ok_t = (xk_t, yk_t, bk_t, hk_t)$ indicates the position of $k \epsilon k1, k2 \ldots\ldots kn$ object in frame Ft. As a result, the object motions can be simply learned by calculating the difference between O_{t-1} and O_t.

4.2.1.2 Re-identification of objects

Instead of comparing features between objects, the proposed method considers the spatial features to calculate relative scaling instead of individual object feature extraction. A single neural network performs this operation, Matching is getting with the comparison of size and distance between two bounding boxes of detection and tracked object; the Matching block aims to find the affinity between the objects to distinguish from others. The $\hat{\text{M}}$ matching block aims to fuse the boundary boxes generated from the optical flow and detected boxes by the YOLO

v3 methods, that is one of the popular and effective techniques for object detection. The boundary boxes generated by the optical flow $O_t = (ok1_t, ok2_t, \ldots\ldots ok n_t | 1 \leq k1, k2, \ldots..kn \leq N)$and the object detection $C_t = \{c_t^i\}$|i= 1,2.....m from image-based object detectors. On the other hand, objects from optical flow contain identity (ID) index $(k1; k2; :::; kn|1 \leq k1, k2, k3 \leq N)$ inherited from F_{t-1}, but may have not accurate boundary boxes in frame t. The detected object from the image needs to assign IDs, the allocation of IDs is either considered the IDs of the object of FlowNet or generating a new one. Moreover, the object of the Flow tracker and detection can be matched. Formally, the matching block takes Bbox proposals from both O_t and C_t, and current frame F_t as input. The bounding box size is informative for the relationship among the objects. The match block are grouped as

Relative scale: It is used to calculate the size of the detected boundary box with respect to tracked boundary box. The relative scale is informative for the relationship between the two boundary boxes. Given a couple of bounding boxes, the detected bounding box is represented as (Xci, Yci, Xcj, Ycj) and targeted have (Xoi, Yoi, Xoj, Yoj) and centers are represented as (Xc, Yc) and (Xo, Yo) respectively, along with corresponding areas A and Ao and AI is the image area of image size (B, H). The relative scale for detected boundary boxes can be calculated as:

$$Rs = \frac{Xci}{B}, \frac{Yci}{H}, \frac{Xcj}{B}, \frac{Ycj}{H}, \frac{A}{AI} \tag{4.1}$$

The equation for relative scale is like the regression coefficient of bbox as proposed [200], but the most obvious cues are measured in the area covered by the detected and tracked objects. to calculate the percentage of area matched between these are using IoU that calculated as the area of intersection and area of union using equation 4.2

$$IoU = \frac{Area \quad of \quad Intersections}{Area \quad of \quad Union} \tag{4.2}$$

The ID is considered one with the highest object score and for the unmatched detection, the proposed model repeats this to match with trajectory (trajectories not assigned in current frames). The detection that cannot be matched by this process is initialized as a new trajectory.

4.3 Results and Analysis

The proposed model is implemented with PyTorch in the python framework. Extensively the model has been trained on a benchmark dataset of MOT, including 2DMOT15, MOT16, MOT17, MOT20 and Waymo. The comparison of the proposed model has been conducted with recent MOT state-of-the-art approaches. The performance analysis with the effect of matching techniques and FlowNet considers the influences of object size and visibility on the proposed method.

4.3.1 Experimental Setup

The motion prediction part contains FlowNet-2 [201] and a regression part to predict the object motion. The method uses a pre-trained FlowNet-2 on MPI-Sintel [202] dataset and fixed weights to train the regression part. In the proposed model, the training of FlowNet uses Adam optimizer with an initial learning rate of 0.001 divided by 10 at every 40 epochs. The proposed model trained for 250 epochs for 32 batch sizes. The fusion is improved from the YOLOv3, in which the backbone network is darknet-53. The backbone network is used the pre-trained weights of Darknet to train the complete model with the ground truth bounding boxes from the MOT datasets. In the proposed matching part, the training process uses the stochastic gradient descent (SGD) optimizer and the initial learning rate is set to .001, divided by 10 after every 10 epochs. The model trained on 24 batch sizes for 50 epochs. As described in the algorithm, some parameters are the confidence score and NMS threshold. In particular, the threshold for confidence score = 0.5, NMS threshold = 0.7, for the benchmark datasets in the experiment. The publicly

Algorithm 2: An algorithm for hybrid end-to-end multi-object tracking

Step 1: Apply detection on the first frame C_0. The detected boundary box is passed through the refinement block. The refinement block kills those boundary boxes with a confidence score lesser than the threshold (0.7) and sequentially applies NMS on each frame. To initialize the trajectory set Tr, the refined detection frame C_0^{ref} is used.
Step 2: The refinement of the current frame detection boundary boxes is done as shown in Step-1.
Step 3: The optical flow generated track boundary boxes O_{t-1} (Boundary box assigned trajectory IDs) in previous frame F_{t-1} that are found in Tr. The frame pair (F_{t-1}, F_t) and O_{t-1} are using as input for optical flow block that produces corresponding boundary box O_t in the current frame F_t.
Step 4: The heatmap head is used to find the center of the object (Xc,Yc). It applied on both boundary boxes, tracked objects as well as detected objects.
Step 5: The matching block fuses the boundary boxes O_t and C_t^{ref}. First, the refinement block refined the O_t boundary box in step-1. The fusion block calculates the relative scale between detected and tracked boundary boxes as illustrated in eq- 4.1. The relative scale is formulated to find the maximum likelihood of a couple of correct matches having a relative smaller scale than a wrong couple match in a soft discriminant manner.
Step 6: The boundary box O_t^{trc} are added to the corresponding trajectory in Tr.
Step 7: For the unmatched detection C_t^{umh}, the model repeat this to match with trajectory Trumh (trajectories not assigned in current frames).
Step 8: The detection that cannot be matched by this process $C_t^{umh'}$ is initialized as a new trajectory.

available datasets for object tracking methods such as CityPerson and ETH have only box annotations that are basically for detection. The MOT17, CalTech PRW and CUHKSYSU datasets have identity annotations and boxes suitable for detection and re-identification. The MOT16 has some part of ETH in its testing set, but for a fair comparison, which is removed from the training set. The overall training strategy is described in 3.4, which is like [167]. To train the proposed method, the MOT20 dataset has both boxes and annotations.

4.3.2 Loss function

The loss function of object motion is:

$$L_{motion} = L1(\Delta Ot, \Delta Ot^*) = ||\Delta Ot - \Delta Ot^*||^2 \tag{4.3}$$

Where Ot* = Ot - Ot-1 described the motion vector of the ground-truth for all objects between a couple of frames F_{t-1} and F_t and O_t indicates motion vector of the output. The mean squared loss function is calculated for Loss regression

$$L_{regression} = \frac{1}{N}\sum_{i=1}^{N} loss_t = \frac{1}{N}\sum_{i=1}^{N} \|Y_{pred} - Y_{GT}\|^2 \tag{4.4}$$

The total loss function is combined as

$$L_{Total} = L_{motion} + L_{regression} \tag{4.5}$$

The performance of the model on different datasets are illustrated in Table-4.1, Table-4.2, Table-4.3, Table-4.4 and Table- 4.5 respectively. the proposed method achieves the almost best performance is mostly tracked objects (MT), the most lost objects (ML) and fewer ID switches compare to existing state-of-the-art approaches.

The result on the waymo test dataset outperforms all existing state-of-the-art methods. The obtained result is 53.29 % MOTA which represents the accuracy percentage is more than 2.11 from quasi-dense, 2.28 from HorizonMOT, and 3.07 from CascadeRCNN-SORTv2. The superior performance of proposed methods without specific training or optimization demands more thorough analysis without sophisticated tracking methods. Finally, To analyze the specific performance of the proposed model in more detail. The flownet-2 calculates every movement of the pixel based on the frame, while the performance is affected in the case of occlusion and object sizes. When an occlusion occurs in the current frame, then the matching may not perform between the pixels of frames. Moreover, optical flow computation

TABLE 4.1: Performance Comparison between the proposed method and other latest track association MOT approaches on the 2DMOT15 Dataset. The last column has frames/second that measure the speed of the model

Tracker	**MOTA**	**IDF1**	**MT**	**ML**	**IDs**	**FPS**
INRALA [203]	34.7	42.1	12.5	30	1112	-
EAMTT [204]	53	54	35.90%	19.60%	7538	<4.0
HybridDAT [205]	35	47.7	11.4	42.2	358	-
AMIR15 [206]	37.6	46	15.8	26.8	1026	-
MDP_SubCNN [207]	47.5	55.7	30.00%	18.60%	628	<1.7
STRN [208]	38.1	46.6	11.5	33.41	1033	-
KCF [209]	38.9	44.5	16.6	31.5	720	-
Tracktor [210]	44.1	46.7	18	26.2	1318	-
CDA_DDAL [211]	51.3	54.1	36.30%	22.20%	544	<1.2
AP_HWDPL [212]	53	52.2	29.10%	20.00%	708	6.7
RAR15 [213]	56.5	61.3	45.10%	14.60%	428	<3.4
TubeTK [214]	58.4	53.1	39.30%	18.00%	854	5.8
FairMOT [215]	60.6	64.7	47.60%	11.00%	591	30.5
Proposed model	61.5	65	49.86%	10.50%	549	37.2

is considered average error over the pixels, so in the case of the small object, the optical flow is probably less accurate, affecting the motion estimation of objects. The Fig-4.2 illustrated the effects of object size and object visibility in the following experiments and graphs to analyze the model performance on the Waymo testing set.

The ratio of non-overlapped areas and their boundary box area is described as the object visibility. As shown in Fig- 4.2 a large number of objects are missing at low visibility. At high visibility, the number of correctly tracked objects increases. The model has gotten tracked object ratio to become steady at the 0:6 object visibility and objects with greater than 0:6 visibility and consider that to analyze the object size. The height of the boundary box is used to represent the size of an object. The performance is perfect at 120 object size (height of boundary boxes).

Table 4.2: Performance Comparison between the proposed method and other latest track association MOT approaches on the MOT16 Dataset. The last column has frames/second that measure the speed of the model

Tracker	**MOTA**	**IDF1**	**MT**	**ML**	**IDs**	**FPS**
EAMTT [204]	52.5	53.3	19.90%	34.90%	910	<5.5
SORTwHPD16 [216]	59.8	53.8	25.40%	22.70%	1423	<8.6
DeepSort_2 [217]	61.4	62.2	32.80%	18.20%	781	<6.4
RAR16wVGG [213]	63	63.8	39.90%	22.10%	482	<1.4
Vmaxx [218]	62.6	49.2	32.70%	21.10%	1389	<3.9
TubeTK [214]	64	59.4	33.50%	19.40%	1117	1
JDE [219]	64.4	55.8	35.40%	20.00%	1544	18.5
LSSTO [220]	49.2	56.5	13.40%	41.40%	606	-
Tracktor [210]	54.4	52.5	19.00%	36.90%	682	-
TAP [221]	64.8	73.5	38.50%	21.60%	571	<8.0
CNNMTT [221]	65.2	62.2	32.40%	21.30%	946	<5.3
POI [222]	66.1	65.1	34%	20.80%	805	<5.0
CTrackerV1 [223]	67.6	57.2	32.90%	23.10%	1897	6.8
FairMOT [215]	74.9	72.8	44.70%	15.90%	1074	25.9
Proposed model	75.0	73.2	45.20%	15.00%	1005	32

Table 4.3: Performance Comparison between the proposed method and other latest track association MOT approaches on the MOT17 Dataset. The last column has frames/second that measure the speed of the model

Tracker	**MOTA**	**IDF1**	**MT**	**ML**	**IDs**	**FPS**
SST [224]	52.4	49.5	21.40%	30.70%	8431	<3.9
TubeTK [214]	63	58.6	31.20%	19.90%	4137	3
CTrackerV1 [223]	66.6	57.4	32.20%	24.20%	5529	6.8
LSSTO [220]	52.7	57.9	17.90%	36.60%	2167	-
CenterTrack [225]	67.3	59.9	19.50%	34.90%	2898	22
Tracktor [210]	53.5	52.3	19.50%	36.60%	2072	-
FairMOT [215]	73.7	72.3	43.20%	17.30%	3303	25.9
Proposed Model	74.2	73.1	44.90%	17.10%	3250	29.9

TABLE 4.4: Performance Comparison between the proposed method and other latest track association MOT approaches on the MOT20 Dataset. The last column has frames/second that measure the speed of the model

Tracker	MOTA	IDF1	MT	ML	IDs	FPS
FairMOT [215]	61.8	67.3	68.80%	7.60%	5243	13.2
Proposed Model	62.0	68.9	70.20%	7.50%	4356	31.5

TABLE 4.5: Performance Comparison between the proposed method and other latest track association MOT approaches on the Waymo dataset

Method	MOTA
Tracktor++	34.8
RetinaTrack	44.92
CascadeRCNN-SORTv2	50.22
HorizonMOT	51.01
Quasi-Dense	51.18
Proposed Method	53.29

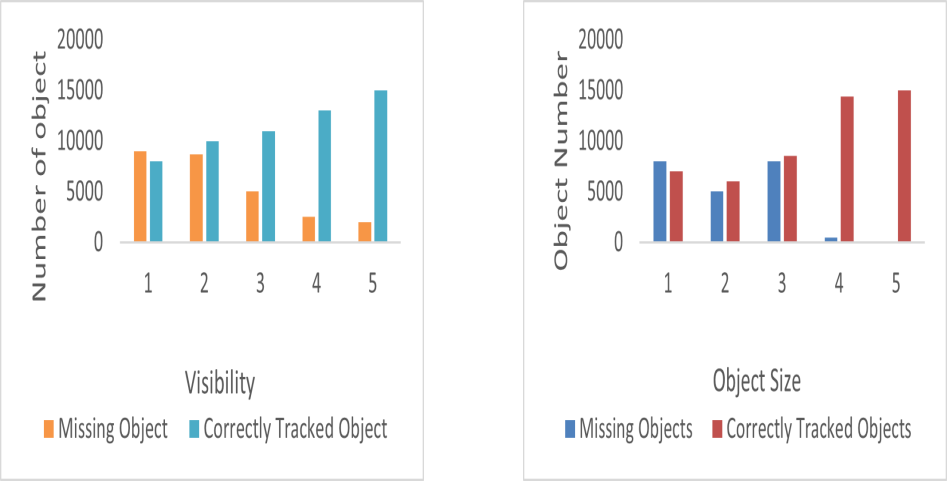

FIGURE 4.2: Effect of visibility (a) and size of the object (b) of the proposed model

FIGURE 4.3: Qualitative Results of the proposed model on Waymo and MOT datasets

4.3.3 Ablation study

For the proposed model, better detection would be the reason for better tracking results. The tracking performance depends on the detected object; thus, the backbone network for the detection of the objects of consecutive frames should be high. To check the generality of the proposed model with different detection method and confidence scores via different backbones, The model is trained with the different backbone networks. The model has achieved 75.0 MOTA when only using the MOT17 dataset for training. The Darknet-19 network provides more speed for the real-time existence of the model compared to ResNet [28] and Darknet-53, while the accuracy of the Darknet-53 is greater than the other backbone.

TABLE 4.6: Performance of the proposed model with the different backbone network through ablation study

Backbone Network	**MOTA**	**FPS**
ResNet-50	65.9	29
ResNet-101	69.7	25
ResNet-152	71.5	22
Darknet-19	67	40
Darknet-53	75	32

4.4 Conclusion

Start by observing why the existing single-shot methods fail to find a prominent result. The proposed model observed that occlusion or object size is challenging for tracking objects. The used detection YOLOv3 is capable of detecting the occluded object with multi-scale detection and the relative scale is responsible for the interrelationship between the boundary boxes of the small object too. The proposed model has computed the relative scale that measures the covered area by the detected and tracked object boundary boxes. The occupied area between these two is computed with IoU and compared with the threshold value that defined the same identifications. The proposed method implements an end-to-end deep-learning approach for solving these issues by avoiding pair-wise appearance-based Re-ID, which can be the costly and indispensable components of most existing MOT approaches. This is a hope that established work is a new tracking paradigm, utilizing the less number parameter and low cost. A compact model can be designed to provide more accurate results with fewer parameters in the future. In the future, a graph neural network-based model can be designed to improve performance and quickly provide the interaction between objects.