TRDP-I: Multi-Object Tracking in Video Sequences

Kush Gupta and Sergio Avello Largo
Proponent: Juan Carlos San Miguel
Supervisors: Csaba Benedek and Jenny Benois-Pineau

***Abstract*— Multi-object tracking (MOT) is defined as the analysis of video sequences in order to establish the location of different objects over a sequence of frames. Multi-object tracking in video is one of the most classical computer vision tasks and the different challenges that it faces could be variations in appearance, clutter, change of illumination, sensor noise, occlusions... It is a key topic regarding related different fields such as video surveillance, virtual-reality gaming or autonomous driving. Although this technology is currently deeply embedded in our daily life, it is still a great unknown for the general public due to its high complexity. Therefore, the main goal of this project is to study in detail several recent state-of-the-art deep learning based multi-object tracking algorithms and then provide a general overview about the most significant aspects of them. The main stages of some of the Multi-object tracking approaches will be discussed and the purpose of each design will be explained. Finally, an experimental study is carried out on different test video sequences from the MOT challenges using one of the deep learning based algorithms presented for the sake of analysing the theoretical concepts introduced throughout the whole document and to obtain some visual results that demonstrates the potential of Multi-object tracking.**

I. INTRODUCTION

Multi-Object Tracking (MOT) is one of the most popular computer vision tasks. Its main goal is to inspect video sequences in order to identify objects belonging to one or more classes, such as people, animals, cars, and many more different objects and then to track them, without any prior information about the number of targets and how they look like. Similarly to object detection algorithms, the output of MOT algorithms is a set of bounding boxes defined by the coordinates of a principal point (center, corners...), its height and its width. In addition, MOT algorithms associate a target ID to each detection for distinguishing among intra-class objects. The vast majority of MOT algorithms share these four steps: Object detection, feature extraction, affinity computation and association.

Lately, the spectacular power of deep learning to represent any type of input as different features has amazed the technological world. Obviously, the computer vision community has quickly adapted to this game changer and thus, in the recent years, most of the top performances in the MOT tasks have been achieved using deep learning based algorithms [1].

Since the potential of this technology is breathtaking, a large variety of approaches have been developed by the research community in order to achieve better results for the MOT tasks. Therefore, the aim of this report is to focus on some of the most interesting state-of-the-art approaches and provide a general overview about their design, mathematical concepts used and results.

This report is organized in eight sections: Introduction, Preliminaries and tools, Related work, Selected approaches, Discussion, Experimental study, Future work and Conclusions.

II. PRELIMINARIES AND TOOLS

This section introduces a brief explanation of the main concepts/tools that should be known in order to fully understand the algorithms and approaches that will be discussed in the following sections.

A. Convolutional Neural Networks (CNN)

A CNN is a type of neural network that allows to process data with high dimensionality (such as images or videos). The basic operation of these networks is carried out in the convolutional layers, where different kernels are applied to the input data in order to extract specific features that may be interesting for future tasks [2].

B. Siamese CNN

A Siamese CNN is a network architecture built using two or more identical (twin) convolutional networks (therefore it needs at least two inputs). These twin networks have the same parameters and same configuration. When training siamese networks it is needed to use two images that belong to the same class (positive pairs) and two images that belong to different classes (negative pairs) [3].

C. Multilayer Perceptron (MLP)

A MLP consists of a system of simple interconnected nodes (neurons) that represents a nonlinear mapping between an input vector and an output vector. The nodes are connected by weights and the output signals which are a function of the sum of the inputs to the node modified by a simple nonlinear function (activation function)[4].

D. Long Short Term Memory (LSTM) and Recurrent Neural Network (RNN)

A LSTM is a unit whose aim is to avoid long-term dependency problems. The design is based on a three gate mechanism (input gate, forget gate, and output gate). The combination of the information from these three gates yields the output of the LSTM unit. The LSTM unit is the core of the RNNs which are a type of artificial neural network designed to recognize patterns in sequences of data by exploiting the temporal dimension [5].

E. Transformer

A Transformer is an state-of-the-art architecture that eschews recurrence and instead of entirely relying on an attention mechanism to draw global dependencies between input and output, it employs an encoder and decoder strategy, but removing recurrence in order to allow significantly more parallelization than methods like RNNs and CNNs [6].

F. Message Passing Network (MPN)

MPNs operate by propagating the features on a graph by exchanging information between adjacent nodes. A typical MPN architecture comprises several propagation layers, where each node is updated based on the aggregation of its neighbour features [7].

G. YOLOv5

YOLOv5 is the latest version of the YOLO object detection algorithms. Apart from dividing the input image into regions and predicting bounding boxes and probabilities for different objects in each region, it integrates adaptive anchor frame calculation on the input, so that it can automatically set the initial anchor frame size in order to adapt to different inputs or datasets.

III. RELATED WORK

This section presents a general overview over the wide range of state-of-the-art techniques for multi-object tracking in video sequences. Some of the best performing MOT methods such as Bergmann et al. [8], Yu et al. [9], Zhou et al. [10], Wang et al. [11], Voigtlaender et al. [12], Zhang et al. [13] employ the standard approach tracking by detection paradigm, which first detects objects in each frame and then associate them over time. These works usually treat re-ID as a secondary task whose accuracy is heavily affected by the primary detection task. As a result, the network is biased to the primary detection task which is not fair to the re-ID task.

Regarding the main steps of the MOT task (mentioned in the introduction), the detection and features extraction ones are widely known concepts that have already been solved with astonishing performances in other computer vision problems. Therefore, few novelties in relation with these stages have been presented in recent years. Nevertheless, going in detail into the affinity stage, an interesting approach was introduced in [14], where Ma et al. decided to directly use the output of a Siamese CNN as an affinity result, instead of employing classical distances between feature vectors. Focusing on improving the association task has become very popular too. Milan et al. [15] used a RNN to predict the probability of existence of a track in each frame or Kieritz et al. [16] used a MLP with two hidden layers to compute track confidence scores. In recent years, some out of the box ideas were presented as well like [7] that instead of relying on the tracking by detection approach, it defines a fully differentiable framework based on Message Passing Networks or [17] that consists of a single and task-agnostic appearance model, which can be learned in a supervised or self-supervised fashion.

IV. SELECTED APPROACHES

In addition to all the methodologies mentioned before, the following approaches were chosen to be discussed in detail due to their significant performance on the different MOT challenges and the interesting concepts proposed by them.

A. FairMOT

In this sub section, we present the technical details of FairMOT [18] regarding its backbone network, the object detection branch and the re-ID branch. FairMOT deals at the same time with the detection and the re-ID tasks instead of detecting first and then the re-ID. This approach uses a anchor less object detection methods, implemented in an anchor-free style which estimates the object centers and sizes represented as position-aware measurement maps. The re-ID branch estimates a re-ID feature for each of the pixel to characterize the object centered at the pixel. A general overview of a one-shot FairMot tracker can be seen in Figure 1.

1) Backbone Network: FairMOT uses ResNet-34 as the backbone in order to strike a good balance between the accuracy and the speed. An enhanced version of Deep Layer Aggregation (DLA) [19] is applied to the backbone to fuse multi-layer features. In addition, convolution layers in all up-sampling modules are replaced by deformable convolutions such that they can dynamically adjust the receptive field according to object scales and poses. These modifications helped to alleviate the alignment issue. The resulting model is named DLA-34. Since the size of input image is Himage × Wimage, the output feature map has the shape of C × H × W where H = $\mathrm{H_{image}}/4$ and W = $\mathrm{W_{image}}/4$.

2) Detection Branch: The detection branch is built on top of CenterNet [19]. In particular, three parallel heads are appended to DLA-34 to estimate heatmaps, object center offsets and bounding box sizes, respectively. Each head is implemented by applying a 3 × 3 convolution (with 256 channels) to the output features of DLA-34, followed by a 1 × 1 convolutional layer which generates the final targets. These three heads will be reviewed next.

2.1 Heatmap Head

This head is responsible for estimating the locations of the object centers. The heatmap based representation, which is the de facto standard for the landmark point estimation task, is adopted here. In particular, the dimension of the heatmap is 1 × H × W. The response in a certain location in the heatmap is expected to be one if it collapses with the ground-truth object center. The response decays exponentially as the distance between the heatmap location and the object center increases. The loss function is defined as pixelwise logistic regression with focal loss [20], where $\hat{\mathbf{M}}$ is the estimated heatmap, and α, β are the predetermined parameters in focal loss. The estimated heatmap loss can be calculated using the equation given below:

$$L_{\text{heat}} = -\frac{1}{N}\sum_{xy}\begin{cases}(1-\hat{M}_{xy})^{\alpha}\log(\hat{M}_{xy}), & M_{xy}=1;\\ (1-M_{xy})^{\beta}(\hat{M}_{xy})^{\alpha}\log(1-\hat{M}_{xy}) & \text{otherwise,}\end{cases}$$

2.2 Box Offset and size Head

The box offset head aims to localize objects more precisely. Since the stride of the final feature map is four, it will introduce quantization errors up to four pixels. This branch estimates a continuous offset relative to the object center for each pixel in order to mitigate the impact of down-sampling. On the other hand, the box size head is responsible for estimating height and width of the target box at each location. The size of the output head can denoted as $\hat{S} \in \mathrm{R}^{2XHXW}$. For each ground (GT) box $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, the size is computed as $s_i = (x_2^i - x_1^i, y_2^i - y_1^i)$.

3) Re-ID Branch: The main goal of the Re-ID branch is to generate features that can distinguish objects. Ideally, affinity among different objects should be smaller than between same objects. To achieve this, a convolution layer with 128 kernels on top of backbone features is applied to extract re-ID features for each location. The resulting feature map is denoted as $E \in \mathrm{R}^{128XHXW}$. The re-ID feature $E_{x,y} \in \mathrm{R}^{128}$ of an object centered at (x, y) can be extracted from the feature map.

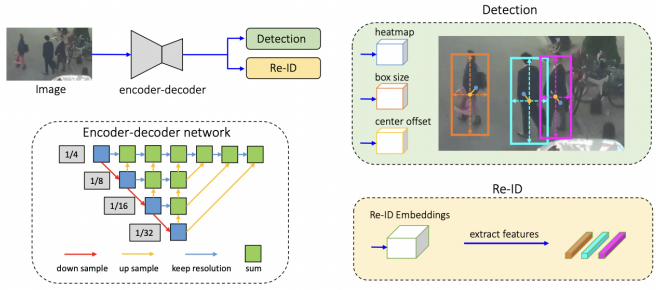

Fig. 1: Overview of one-shot tracker FairMOT. The input image is first fed to an encoder-decoder network to extract high resolution feature maps (stride = 4). Then, two homogeneous branches were added for detecting objects and extracting re-ID features, respectively. The features at the predicted object centers are used for tracking. (Image source[18])

4) Association: For the association stage, FairMOT firstly follows the well-known DeepSORT [21] algorithm (explained later in this section). As a second stage, the unmatched detections and tracklets are matched according to the overlap between their boxes. In particular, the matching threshold $\tau 2$ was set to 0.5. The appearance features of the tracklets are updated each time step to handle appearance variations [22][23]. Finally, the unmatched detections are initialized as new tracks and the unmatched tracklets are saved for 30 frames in order to deal with future reappearance.

B. TRANSCENTER

In this subsection, we will discuss the technical details of another state of the art approach, TransCenter [24]. It is based on an encoder-decoder structure where the encoder extracts the image information and the decoder finds the best correlation between the object query and the encoded image features with an attention module. The attention module transforms the inputs into Query (Q), Key (K), and Value (V) with fully-connected layers. Having Q, K, V, the attended features are calculated with the attention function (equation below), where h is the hidden dimension of Q, K, and V.

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{h}})V$$

TransCenter proposes to tackle the MOT task by dedicating the TransCenter Decoder for the two main tasks: detection and temporal association. This approach is different from other transformer-based MOT methods, as it questions the use of sparse queries without positional correlations (i.e. noise initialized), and explores the use of image-related dense queries producing dense representations within transformers. To that aim, it deploys a query learning networks (QLN) that is responsible for converting the output of the encoder into the input of the decoder.

A generic pipeline of TransCenter is illustrated in Figure 2. The RGB images at the t and t-1 time steps are given as inputs to the weight-shared Transformer encoder and it generates the dense multi-scale attended features called memories M_t and M_{t-1}, respectively. They are the inputs of the **QLN**. The latter produces two sets of output pairs, a tracking query **(DQ)** and the memory **(DM)** for detecting the objects at time t, and on the other hand a tracking query **(TQ)** and the memory **(TM)** for associating the objects at time step t with those from previous time step t-1. Furthermore, the TransCenter Decoder, leveraging the deformable transformer [25], is used to correlate the detection/tracking queries with the memories. In the next step, **TQ** interacts with **TM** in the cross-attention module of the TransCenter Decoder, resulting in the tracking features **(TF)**. Similarly, the detection features **(DF)** are the output of the cross-attention between **DQ** and **DM**. To produce the output dense representations, **DF** is used to estimate the object size S_t and the center of thr heatmap C_t. **TF** is used to estimate the tracking displacement T_t [24].

Brief discussion on the main units of the TransCenter method: The TransCenter Decoder, the Multi Scale Queries and the output branches.

1) TransCenter Decoder: TransCenter Decoder handles detection and tracking in a parallel. It uses a Tracking Deformable Cross-Attention module (TDCA) to combine the information from **TQ** and **TM** and at the same time a Detection Deformable Cross-Attention module (DDCA) associates **DQ** and **DM**.

The computational cost of all this processing is fairly high.

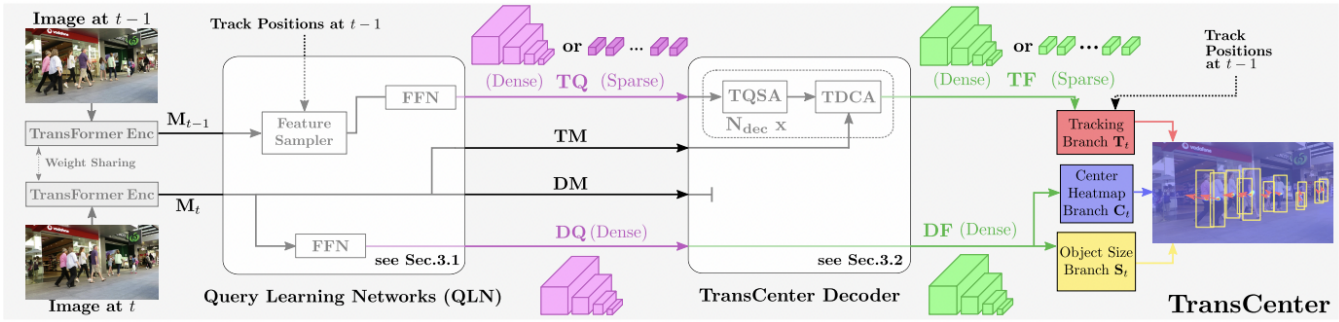

Fig. 2: General overview of a TransCenter pipeline. (Image source [24])

2) Pixel-level Dense Multi Scale Queries: TransCenter uses dense pixel-level queries without overlapping to produce dense heatmaps that infer the probability of having a person's center at a given pixel coordinate. Recalling, that such queries are multi-scale obtained from the multi-scale M_t, via a first QLN obtaining DQ_t. Then, it uses two different queries for the dual decoder: a second QLN processes DQ_t to obtain TQ_t. Queries are multi-scale and exploit the multi-resolution structure of the encoder, allowing for very small targets to be captured by those queries. Dense queries also make the network more flexible since it can adapt automatically to arbitrary image size without re-parameterizing.

3) The Center, The Size and the Tracking Branches: After obtaining **DF** and **TF** as outputs from the TransCenter Decoder, TransCenter employs different branches to output the object center heatmap C_t, its bounding box size S_t as well as the tracking displacements T_t. **DF** contains feature maps of four different resolutions, namely 1/32, 1/16, 1/8, and 1/4 of the input image resolution. For the center heatmap and the object size, the feature maps at different resolutions are combined using deformable convolutions [25] and bilinear interpolation, following the architecture shown in Figure 3, into a feature map of 1/4 of the input resolution, and finally into $C_t \in [0,1]^{H/4xW/4}$ (H and W are the input image height and width, respectively) and $S_t \in \mathbb{R}^{H/4xW/4x2}$ (the two channels of S_t encode the object width and height). Regarding the tracking branch, the tracking features **TF** are sparse of size depending on the number of tracks at t-1 where one tracking query feature corresponds to one track at t-1. **TF**, together with object positions at t-1 (sparse TQ) or center heatmap C_{t1} and **DF** (dense TQ), are input to two fully-connected layers with ReLU activation. They predict the horizontal and vertical displacements T_t of tracks t-1 in the adjacent frames.

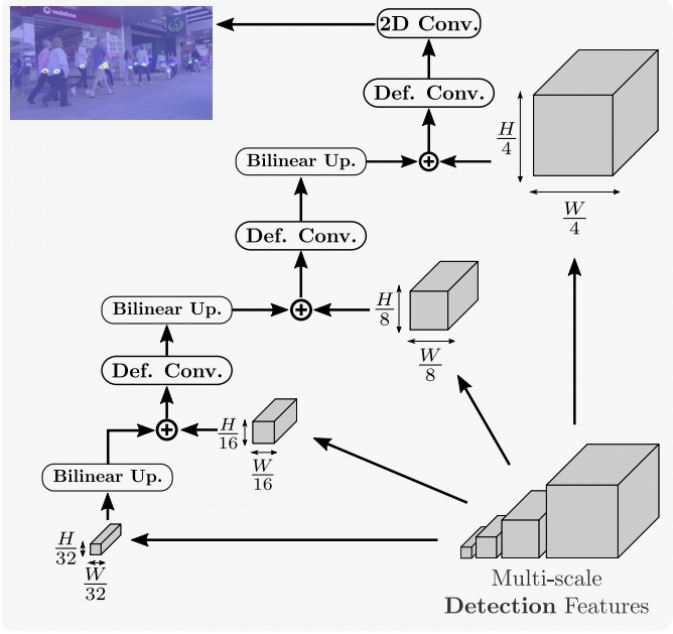

Fig. 3: Architecture of the center heatmap branch. (Image source [24])

C. DeepSORT

DeepSORT [21] is one of most popular and widely used object tracking frameworks. Although it was presented five years ago, its performance is still fairly remarkable.

DeepSort is based on a Kalman filter [26] to predict tracklet locations in the following frame. The Kalman filter algorithm assumes a simple linear velocity model defined on an eight-dimensional state space (u, v, a, h, u', v', a', h'), where (u, v), are the coordinates of the center of the bounding box, a is its aspect ratio and h is its height. The remaining variables are the corresponding velocities of the previous variables.

For each of the detections provided as inputs to DeepSORT, a so called 'track' is created. This track has a parameter that allows to eliminate or not a track depending on on how long ago was the last successful detection. Furthermore, during the first frames, there is a threshold that marks the minimum number of detections.

Once the Kalman filter has generated the prediction of the variables that define the bounding boxes, it's time to associate them with new detections. For this task a distance metric that measures the quality of the association and an efficient algorithm to associate the data are needed.

The squared Mahalanobis distance [27] between the predicted and detected boxes was proposed by the DeepSORT

authors to incorporate the uncertainties from the Kalman filter. This metric is really effective when measuring distances between two distributions (like the variables under the Kalman filter). It is also interesting to mention that in modern versions of DeepSORT, a metric based on a combination of the Mahalanobis distance with the cosine distance computed on re-ID features is introduced because it can outperform the squared Mahalanobis distance in certain complex scenarios.

Finally, in order to complete the assignment tasks the Hungarian algorithm [28] is applied using the computed distances as assignment costs.

V. DISCUSSION

The basic pipeline for multi-object tracking consists of various blocks, like object detection, feature extraction, affinity computation and association. The standard approaches are mainly based for the objects using the Kalman filter [26] for predicting the position of different objects, and using the Hungarian algorithm [28] to solve the association problem. However, despite the effectiveness of the Kalman filter, it fails in many real-world scenarios like occlusions, different viewpoints, etc. Hence, in order to overcome these challenges, in the recent years, deep learning has been widely applied along in these blocks.

Along this project several state-of-the art deep learning approaches have been presented in order to demonstrate the enormous potential of the combination of multi-object tracking. However, DeepSORT [29], which is an early deep learning-based tracking method, is still one of the most widely used methods. Hence, in order to gain a better and deeper understanding of the fundamentals of the multi-object tracking paradigm, DeepSORT was selected for a further analysis the with experimentation in the following section.

VI. EXPERIMENTAL STUDY

In this section, various experiments are performed using YOLOv5 (as a deep learning object detector) and DeepSORT following method proposed in [30]. The results obtained on different test video sequences will be discussed in the following subsections. These different test video sequences belonged to different MOT challenge sequences (MOT15, MOT16, MOT19, MOT20).

A. DATASET

In order to perform the experimental study, five different test video sequences were selected. Each of them, have a different level of complexity and challenges.

1) MOT15-TUD crossing: The video 'TUD crossing' [31] was recorded from a side view. The scene consists of people crossing a road at the signal. It was recorded in day light with 25 frames per second. The video was recorded in 2008, and the resolution of the camera used was 640x480. It has a total of 201 frames. Even though, the resolution is not ideal, the people and the cars in the background can be identified clearly.

2) MOT16-Pedestrian street night: The video named 'Pedestrian street night' [32] was recorded from an elevated viewpoint. The video scene consists of a view of a pedestrian street at night, recorded at a resolution of 1920x1080, with 30 frames per second. It has 1050 number of frames. Since the scene was recorded from an elevated viewpoint, sometimes the objects are partially/fully occluded at the frame edges.

3) MOTS20- ADL-Rundle-1 : The video 'ADL-Rundle-1' [33] was filmed in a busy pedestrian street at eye level by a moving camera. It was filmed with a resolution of 1920x1080, with 30 frames per second (FPS). It has 500 frames. Since the video was filmed with a moving camera the video has jitter issues.

4) MOT20- Crowded indoor train station: The video 'Crowded indoor train station' [34] was recorded at a crowded indoor train station, with a of resolution of 1920x1080 and 25 frames per second. As the video was recorded indoor, there was a big display with some videos (advertisements) playing on it. Due to which, there are light reflections from the floor causing illumination changes in the upper part of the frame.

5) MOT-KITTI-19: The video 'KITTI-19' [35] contains a street scene filmed with a camera mounted on a moving vehicle. It was recorded at 10 frames per second and a resolution of 1238x374. As the video was recorded from a camera mounted on a moving moving vehicle, there is a lot of illumination changes and jitter. Due to sudden changes in illumination, sometimes, it was difficult to detect and track an object.

B. STUDIED PARAMETERS

In order to establish a better understanding of the DeepSORT tracker, an experimental study, was performed. The tracker [30] was tested with five different configurations for the below mentioned parameters for all the test video sequences.

Max_dist:- A distance metric for measurement-to-track association.

Max_IOU_dist:- Maximum over-lap between the ground truth and predicted bounding box.

Max_Age:- Maximum number of missed misses before a track is deleted.

For detection, the pretrained YOLOv5 model 'crowdhuman_yolov5m.pt' was used. Table I below shows the different set of parameters used for each run.

C. Results

The visual results obtained, for the performed experiments were shown in Figures 4, 6, 7, 8. In the results, it can observed that a few times due to missed detections the tracker is unable to track the objects. Apart from the missed detections, it works almost perfectly for all different configurations and at all the different levels of complexity i.e (crowded

Runs	DeepSORT Model	Max_dist	Max_IOU_dist	Max age
1^{st}	osnet_x0_25 _market1501	0.2	0.7	70
2^{nd}	osnet_x0_25 _market1501	0.3	0.6	60
3^{rd}	osnet_x0_75 _market1501	0.4	0.6	50
4^{th}	osnet_x0_75 _market1501	0.4	0.5	50
5^{th}	osnet_x0_75 _market1501	0.4	0.6	60

TABLE I: Different set of parameters used for DeepSORT, for 5 different runs. All the five configurations were tested on all five test videos sequences.

scenarios, sudden illumination changes, bad illumination, elevated view point, etc).

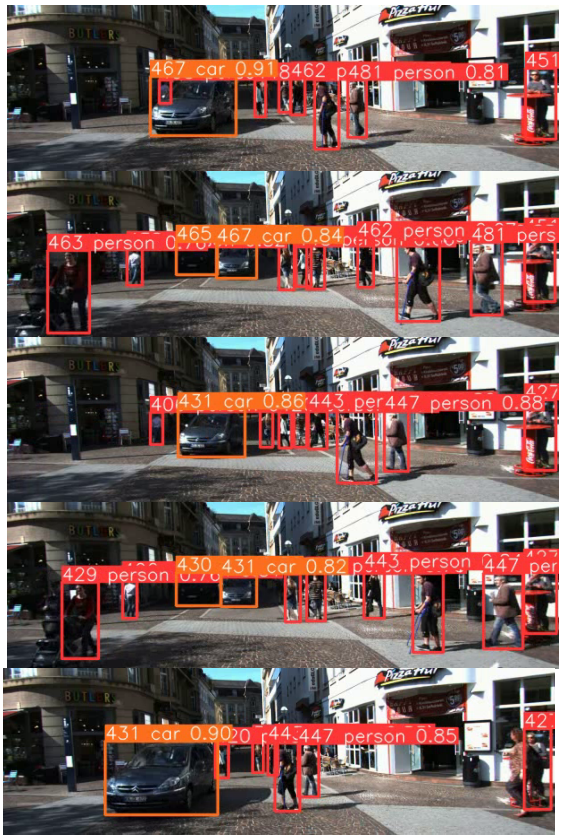

Fig. 4: Tracking results obtained for the video sequence,'KITTI'. Each row shows the results of sampled frames in the chronological order of the test runs. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities.

VII. FUTURE WORK

After reviewing all the concepts and methodologies explained along this report, it is crystal clear that the TransCenter approach is one of the most challenging state-of-the-art methods. It would be really interesting to further research on the mathematical background behind it like, transformers and try to apply them to other approaches which had obtained good results.

On the other hand, since the non deep learning based approach like Kalman filter is still working in an impressive way, it could be attractive to apply the actual knowledge about multi-object tracking for developing (or at least try) a game-changer model not based on deep learning.

VIII. CONCLUSIONS

In conclusion, while working on the project we have gone through a variety of deep learning based multi-object tracking approaches like YOLOv5 + DeepSORT, FairMot, MPN... and some state-of-the-art tracking approaches like TransCenter. During the project development we have gained wide knowledge about a lot of new concepts such the use of graph theory, the power of encoder-decoder modules in multi-object tracking and the use of combination of different deep learning techniques to extract valuable features. Furthermore, we have established a better and deeper understanding about the fundamentals for multi-object tracking.

On the other hand, during the working sessions, we have faced several challenges while trying to understand some of the new state-of-the-art concepts that are applied to multi-object tracking. In addition, while studying the TransCenter approach, we tried to set up the environment and run the proposed code. Unfortunately, we were unable to do it due to unavailability of a suitable platform with supporting GPUs, and the lack of time to solve some critical errors that appeared in the code. Nevertheless, during the process we have learned to tackle such difficult scenarios and to find an alternative solution for them. We are glad to have achieved the learning goal defined for this project and we are really satisfied with the knowledge gained during the whole researching process.

REFERENCES

[1] S. Chen, Y. Xu, X. Zhou, and F. Li, "Deep learning for multiple object tracking: A survey," *IET Computer Vision*, vol. 13, 01 2019.

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[3] A. Nandy, S. Haldar, S. Banerjee, and S. Mitra, "A survey on applications of siamese neural networks in computer vision," in *2020 International Conference for Emerging Technology (INCET)*, pp. 1–5, 2020.

[4] M. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14, pp. 2627–2636, 1998.

[5] Z. Yutao, H. Wu, H. Cheng, K. Qi, K. hu, C. Kang, and J. Zheng, "Social graph convolutional lstm for pedestrian trajectory prediction," *IET Intelligent Transport Systems*, vol. 15, 03 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[7] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," *CoRR*, vol. abs/1912.07515, 2019.

[8] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," *CoRR*, vol. abs/1903.05625, 2019.

[9] A. Hornakova, R. Henschel, B. Rosenhahn, and P. Swoboda, "Lifted disjoint paths with application in multiple object tracking," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 4364–4375, PMLR, 13–18 Jul 2020.

[10] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1809–1814, 2018.

[11] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *CoRR*, vol. abs/1909.12605, 2019.

[12] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," *CoRR*, vol. abs/1902.03604, 2019.

[13] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," *CoRR*, vol. abs/2110.06864, 2021.

[14] L. Ma, S. Tang, M. J. Black, and L. Van Gool, "Customized multi-person tracker," in *Computer Vision – ACCV 2018*, Springer International Publishing, Dec. 2018.

[15] A. Milan, S. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)* (S. Singh and S. Markovitch, eds.), (United States of America), pp. 4225–4232, Association for the Advancement of Artificial Intelligence (AAAI), 2017. AAAI Conference on Artificial Intelligence 2017, AAAI 2017 ; Conference date: 04-02-2017 Through 10-02-2017.

[16] H. Kieritz, W. Hubner, and M. Arens, "Joint detection and online multi-object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[17] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. H. S. Torr, and L. Bertinetto, "Do different tracking tasks require different appearance models?," 2021.

[18] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "A simple baseline for multi-object tracking," *CoRR*, vol. abs/2004.01888, 2020.

[19] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019.

[20] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.

[21] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017.

[22] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[24] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense queries for multiple-object tracking," *CoRR*, vol. abs/2103.15145, 2021.

[25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable {detr}: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021.

[26] R. E. Kálmán, "A new approach to linear filtering and prediction problems" transaction of the asme journal of basic," 1960.

[27] G. Mclachlan, "Mahalanobis distance," *Resonance*, vol. 4, pp. 20–26, 06 1999.

[28] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, 05 2012.

[29] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017.

[30] M. Broström, "Real-time multi-camera multi-object tracker using yolov5 and deepsort with osnet." https://github.com/mikel-brostrom/Yolov5_DeepSort_OSNet, 2022.

[31] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," 06 2008.

[32] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016.

[33] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: multi-object tracking and segmentation," *CoRR*, vol. abs/1902.03604, 2019.

[34] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," *CoRR*, vol. abs/2003.09003, 2020.

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Fig. 5: Tracking results obtained for 'Crowded indoor train station'. Each row shows the results of sampled frames in the chronological order of the test runs. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities.

Fig. 6: Tracking results obtained for 'Pedestrian street night'. Each row shows the results of sampled frames in the chronological order of the test runs. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities.

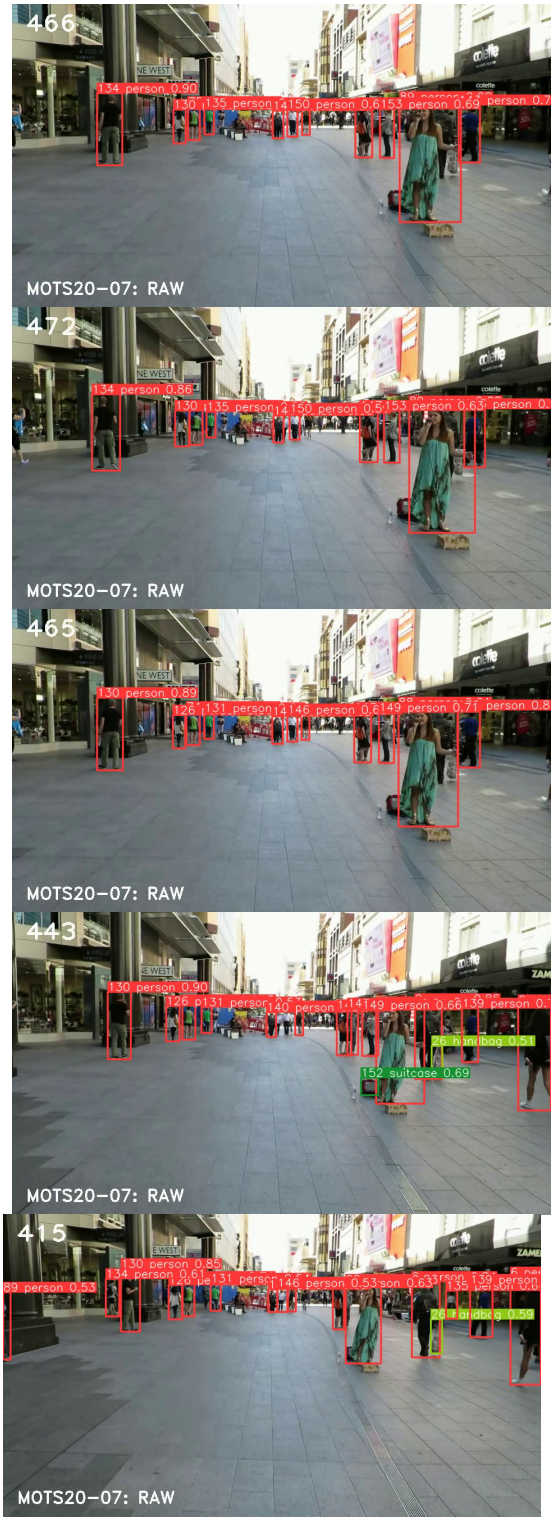

Fig. 7: Tracking results obtained for MOTS20-'ADL-Rundle-1'. Each row shows the results of sampled frames in the chronological order of the test runs. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities.

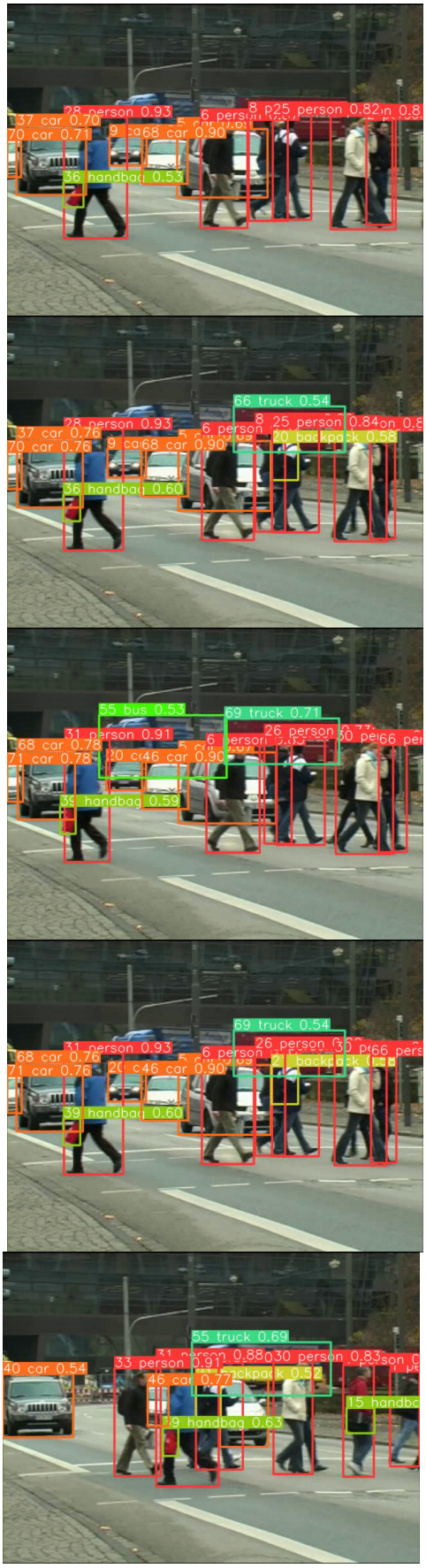

Fig. 8: Tracking results obtained for 'TUD crossing'. Each row shows the results of sampled frames in the chronological order of the test runs. Bounding boxes and identities are marked in the images. Bounding boxes with different colors represent different identities.