TRDP-II: 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 the 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, and occlusions... It is a key topic for different related 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. On TRDP I, a deep analysis of the theoretical and mathematical concepts (CNN, Transformers, Object detectors, Kalman filter...) was carried out in order to fully understand several recent state-of-the-art deep learning-based multi-object tracking algorithms such as FairMot, TransCenter, or DeepSORT. In the end, an experimental study using a model combining the object detector YOLOv5 and the tracker DeepSORT was performed for the sake of demonstrating the potential of Multi-object tracking. On the other hand, for TRDP II, since MOT is a fairly wide concept, we decided to focus our efforts just on pedestrian tracking. In addition, the tracker model from TRDP I (DeepSORT) was updated consequently to a new version called StrongSORT. Finally, the use of a unique synthetic dataset completes the novelties for the second part of the project.**

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 the following 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 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 MOT tasks. Therefore, the aim of this report is to focus on one of the most interesting state-of-the-art approaches: Yolov5 + StrongSORT [2] and combine its astonishing performance with the potential of the synthetic data for providing a general overview of its behavior for pedestrian tracking in complex scenarios.

This report is organized into eight sections: Introduction, Related Work, Selected approaches, Dataset, Experimental Methodology, Results, Conclusions, and Future work.

II. RELATED WORK

This section presents a general overview of 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. [3], Yu et al. [4], Zhou et al. [5], Wang et al. [6], Voigtlaender et al. [7], Zhang et al. [8] 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 toward 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), detection and feature extraction are widely known concepts that have already been solved with astonishing performances in other computer vision problems. Therefore, few novelties in relation to these stages have been presented in recent years. Nevertheless, going into detail about the affinity stage, an interesting approach was introduced in [9], 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. [10] used an RNN to predict the probability of the existence of a track in each frame or Kieritz et al. [11] used an MLP with two hidden layers to compute track confidence scores. In recent years, some out-of-the-box ideas were presented as well like [12] that instead of relying on the tracking by detection approach, defines a fully differentiable framework based on Message Passing Networks or [13] that consists of a single and task-agnostic appearance model, which can be learned in a supervised or self-supervised fashion.

III. SELECTED APPROACHES

In addition to all the methodologies mentioned before and as it was explained in the introduction section, our model

was updated from YOLOv5 + DeepSORT to YOLOv5 + StrongSORT [2], therefore the following methodologies were chosen to be discussed in detail due to their role in our approach.

A. YOLOv5

In this sub-section, we present the technical details of YOLOv5. Due to its speed and accuracy, YOLO [14] is one of the most famous object detection algorithms and it is still widely used in many MOT models. Therefore, there was no need to update it, thus keeping it as the object detector used for both TRDP I and TRDP II.

YOLOv5 is a compound-scaled object detection model trained on the COCO dataset and developed by the company Ultralytics. YOLO is an acronym for 'You only look once' and it is based on a neural network framework called Darknet. The main idea of the YOLO detector is to forward a whole image only once through a single neural network. That is why it is really fast. This network divides the image into regions and predicts bounding boxes and probabilities for different objects in each region. YOLOv5 is one of the latest versions of the YOLO algorithms and 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.

Regarding its architecture, YOLOv5 consists of four main parts: input pre-processing, backbone, neck, and output. The preprocessing of the input data includes mosaic data augmentation [15] and adaptive image filling to enhance the quality of the information provided by the input. The backbone network is based on a cross-stage partial network (CSP) [16], which reduces the computational cost of the process and a spatial pyramid pooling (SPP) [17] that performs feature extraction at different sizes. In the neck network, some concatenations, mixes, and convolutions from different network layers are performed in order to enhance the robustness of the detections. Finally, in the last detection stage, the head output of the neck network is used to predict targets of different sizes on feature maps. This architecture is shown in Figure 1.

It is also important to mention that YOLOv5 includes several versions depending on the number of feature extraction modules and convolution kernels in the network. Some examples are YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x...

B. DeepSORT

DeepSORT [19] is one of the 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 [20] 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

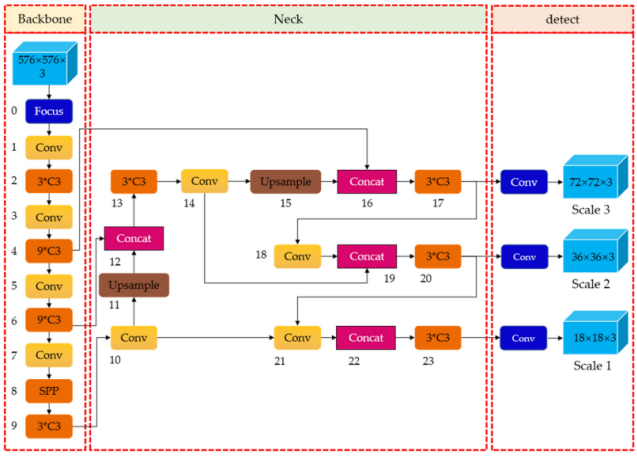

Fig. 1: Network architecture of YOLOv5. (Image source [18])

previous variables. DeepSORT can be defined as a two-branch framework, that is, appearance branch and motion branch, as shown in the top half of Figure

In the appearance branch, for each of the detections provided as inputs to DeepSORT, a so-called 'track' is created. This track has a parameter that allows eliminating or not a track depending 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.

In the motion branch, 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 [21] 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 [22] is applied using the computed distances as assignment costs.

DeepSORT was used as the tracker on the framework developed on TRDP I with really positive results. Nevertheless, during the experiments, we realized that the tracker committed some errors in the association stage while processing scenes with high complexity. Thus, for the second part of the TRDP project, we decided to modify the tracker to an updated version of DeepSORT called StronSORT, which will be introduced in the following subsection applying the concepts explained for DeepSORT.

C. StrongSORT

In this subsection, we will discuss the technical details of StrongSort [23] comparing them with its previous version DeepSORT. The improvements over DeepSORT are mainly carried out in the two branches.

In the appearance branch, the original CNN used in DeepSORT was replaced by a stronger appearance feature extractor named BoT [24] that takes a pre-trained ResNeSt50 [25] on the DukeMTMCreID [26] dataset as the backbone for the sake of extracting much more discriminative features. Furthermore, the appearance state for the i-th tracklet at frame t in an exponential moving average (EMA) follows the strategy proposed in [27]. The exponential moving average updating strategy not only reduces the time consumption but also improves the matching quality.

In the motion branch, the standard Kalman filter of DeepSORT might fail in scenarios with high complexity due to low-quality detections and missing information on the scales of detection noise. To solve this problem, the Noise Scale Adaptive Kalman (NSA Kalman) algorithm which adaptively modulates the noise scale according to the quality of object detection is chosen to replace the original one.

In addition, the assignment problem is solved by combining both appearance and motion branches instead of employing only the appearance feature distance during matching.

Finally, another interesting point to comment on is that the matching cascade in DeepSORT limits its performance as the tracker becomes more powerful. Therefore, the matching cascade is replaced with the vanilla global linear assignment.

All these modifications can be seen in Figure 2.

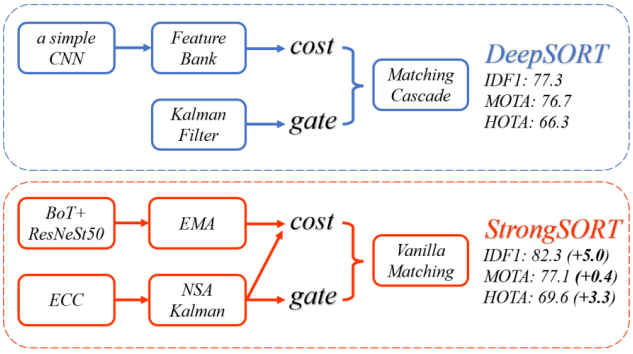

Fig. 2: Differences between DeepSORT and StrongSORT. (Image source [23])

IV. DATASET

In this section, we discuss the MOTsynth dataset used for fine-tuning the detector (YOLOv5) and two different MOT datasets used for the evaluation of the tracking algorithm for the experimental study. Each of the test video sequences has a different level of complexity and challenges.

1) MOTSynth: Often, obtaining a real video sequence could be time-consuming, and annotating the video sequence manually is very costly. Moreover, issues like user privacy and human errors in the annotations are associated with the dataset. Hence, in order to address these issues we replaced the real data with synthetic data known as MOTSynth [28]. It is a large database for pedestrian detection and tracking in urban scenarios which was created by exploiting the highly photorealistic video game Grand Theft Auto V. It is a collection of 764 full-HD videos, each 1800 frames long, recorded at 20 fps. Using synthetic data for model training can be advantageous as the user data privacy concerns can be addressed and human error in the annotation of crowded public environments can be minimized. MOTSynth has a huge variety in terms of environments, camera viewpoints, object textures, lighting conditions, weather, seasonal changes, and object identities which can be observed in Figure 3. In order to obtain diverse actors, it uses generative attributes of 579 pedestrian models, provided by the GTA-V game, e.g., different clothes, backpacks, bags, masks, hair, and beard styles, yielding over 9,519 unique pedestrian identities in total. 256 screenplays were set manually and combined with 128 screenplays from [29] summing to a total of 384 screenplays and each screenplay was recorded twice, one during the day and one during the night, totaling 768 generated diverse sequences. Hence, each video sequence contains 29.5 people per frame on average and a maximum of 125 people. In total there were more than 40M bounding boxes and over 1.3M densely annotated frames.
One of the goals for the TRDP-II is to verify if MOTSynth can be used as a replacement for real data on tasks such as pedestrian detection, re-identification, segmentation, and tracking. Also, through the experimental evaluations, we will try to prove that diversity in the database plays a pivotal role in bridging the synthetic-to-real gap.

2) MOT-16: These video sequences are a benchmark dataset provided by the MOT challenge. They contain unconstrained environments filmed with both static and moving cameras of resolution of 1920x1080 at 30 frames per second. These test video sequences provide a low level of complexity in terms of the number of people in a given frame and also in terms of illumination changes, viewpoint and occlusion. Figure 4 (top) shows the MOT16-02 sequence recorded in the daytime on a street, as it can be noticed that the number of pedestrians in the scene is $<$ 50 (approximately), whereas Figure 4 (bottom) shows the MOT16-10 sequence. It can be observed that even though the sequence was recorded at night time we have enough illumination and also the are fewer people in the given frame.

3) MOT-20: These video sequences are also a benchmark dataset provided in the MOT challenge. They were recorded at crowded indoor/outdoor places, from an elevated fixed point camera of a resolution of 1920x1080 at 25 frames per second. These test video sequences provide a very high level of complexity in terms of the number of people in a given frame and in terms of illumination changes, viewpoint, and occlusion. Figure 5 (top) shows the MOT20-01 sequence recorded inside a train station. As it can be seen that there are

Fig. 3: A glimpse of MOTSynth, a large and diverse dataset for pedestrian detection, re-identification, and multi-object tracking (Image source [28]).

Fig. 4: Shows a subset of the video sequences used for evaluating the algorithm from the MOT-16 dataset.

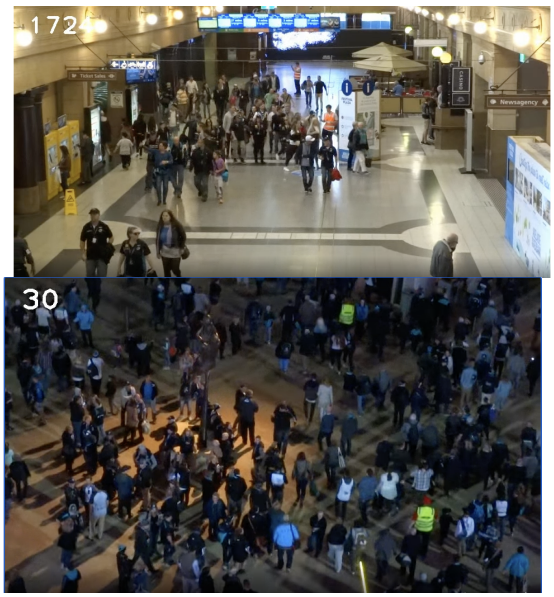

Fig. 5: Shows a subset of the video sequences used for evaluating the algorithm from the MOT-20 dataset.

light reflections from the floor causing illumination changes whereas Figure 5 (bottom) shows the MOT20-05 sequence, as it can be observed that the sequence has poor illumination being night time and also the number of people in the given frame is $>$ 100 (approximately).

V. EXPERIMENTAL METHODOLOGY

In this section, we discuss the experimental methodology used in the TRDP-II. Initially, to establish a baseline we evaluate the algorithm with the pre-trained YOLO weights (crowdhuman [30] and YOLOV5x6). Once the baseline was established we planned to fine-tune the detector using the synthetic data in order to enhance the tracking algorithm. To begin with, we trained the crowdhuman weights using 8990 synthetic images for 6 epochs and validated the model after each epoch using 1798 images. During the training, we could observe that the training loss was decreasing however the validation loss started to increase after a certain point because the model started to overfit the data. Hence, we decided to increase the training data to 14384 images and validation data to 5394 images. After this, we fine-tuned the crowdhuman and the yolov5x6 weights for 30 epochs with SGD loss function and we could see improvement in the final results which are discussed in the Results section.

For analyzing and comparing all these results a novel MOT evaluation metric named HOTA (Higher Order Tracking Accuracy) [31] was chosen, which explicitly balances the effect of performing accurate detection, association, and localization into a single unified metric for comparing trackers. The final HOTA is the geometric mean of the association and detection accuracy averaged over different localization

thresholds which helps us to provide a single score for tracker evaluation that fairly combines all different aspects of tracking evaluation. The range of the values it can present goes from 0 to 100.

To make the MOT task more user-friendly we developed a GUI where the users can select the video sequences and the model used for detection. After selecting suitable values, the GUI shows a display window that visualizes the algorithm output. Figure 6 shows the main landing page of the GUI.

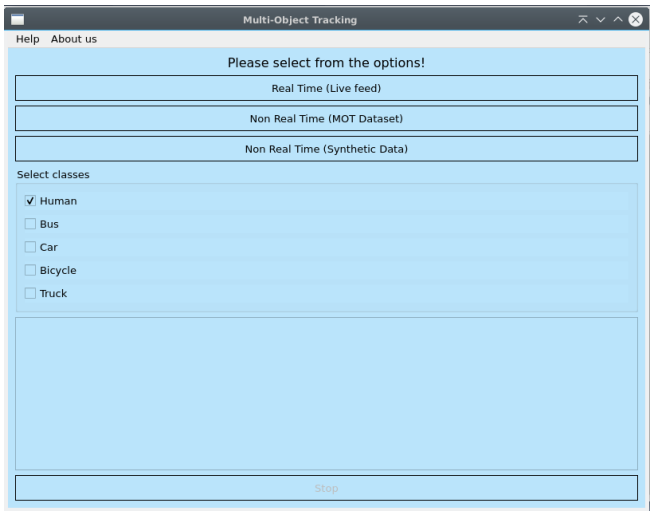

Fig. 6: Main MOT GUI window - Landing page

The sequence of the experiments performed on the test data sequences MOT-16 and MOT-20 is defined below:

- Experiment-1: Establish baseline HOTA for the model using pre-trained YOLO weights (Yolov5x6 and crowdhuman).
- Experiment-2: Evaluate HOTA for the model using the YOLO weights (Yolov5x6 and crowdhuman) fine-tuned on 8990 images.
- Experiment-3: Evaluate HOTA for the model using the YOLO weights (Yolov5x6 and crowdhuman) fine-tuned on 14384 images.

VI. RESULTS

In this section, we discuss the results obtained after performing different experiments mentioned in the section above.

Table I and II shows the HOTA accuracy of the algorithm for various experiments performed on the MOT-16 sequence. In I we compare the baseline model with the fine-tuned yolov5x6 model. Here it can be observed that the model accuracy after fine-tuning decreases because we fine-tuned the model with training images containing crowded scenarios and since MOT-16 sequences are not crowded the model gives false positive detections which accounted for the reduction in the model accuracy. In II when fine-tuning the crowdhuman model, after increasing the training data we can notice an improvement in the HOTA accuracy when comparing experiment 2 and experiment 3. Table III and IV demonstrates the results for MOT20 video sequences. In III we can notice an improvement in the fine-tuned yolov5x6 models when compared to baseline accuracy for crowded scenarios. Similarly, in IV it can be observed that the HOTA accuracy of the fine-tuned model increases as we further increase the size of the training data from experiment 2 to experiment 3.

Video Sequences	Experiment 1 HOTA	Experiment 2 HOTA	Experiment 3 HOTA
Mot16-02	28.843	20.854	22.938
Mot16-09	52.336	35.135	38.632
Mot16-10	37.663	34.445	36.595
Mot16-11	57.335	41.983	44.657
Combined	**41.978**	33.104	34.406

TABLE I: comparing HOTA, the baseline yolov5x6 model with the fine-tuned yolov5x6 model from experiments 2 and 3 on MOT-16 sequences.

Video Sequences	Experiment 1 HOTA	Experiment 2 HOTA	Experiment 3 HOTA
Mot16-02	30.781	24.972	25.537
Mot16-09	50.677	43.119	43.523
Mot16-10	43.324	38.877	39.044
Mot16-11	63.489	51.297	54.147
Combined	**45.793**	38.198	**39.257**

TABLE II: comparing HOTA, the baseline crowdhuman model with the fine-tuned crowdhuman model from experiments 2 and 3 on MOT-16 sequences.

The visual results obtained, after using the fine-tuned weights from the experiment-3 were shown in Figures 7, 8, 9, 10. Figures 7 and 8 shows the detection of MOT16 video sequences using the fine-tuned crowdhuman and yolov5x6 weights respectively. The difference between the two fine-tuned detectors can be observed in the first frame of figures 7, 8. Since in MOT-16 the scenes are not crowded we can say that both detectors are detecting most of the pedestrians in the given frame.

Similarly, Figures 9 and 10 show the detection results on MOT20 video sequences using the fine-tuned crowdhuman and yolov5x6 weights respectively. Since the MOT-20 video sequence is complex and crowded we can observe many missing detections in Figures 9 and 10. As it can be observed from III and IV that for crowded scenarios we can further improve our results by further fine-tuning the detector with a large number of synthetic training images.

Video Sequences	Experiment 1 HOTA	Experiment 2 HOTA	Experiment 3 HOTA
Mot20-01	27.245	34.234	36.398
Mot20-02	20.498	33.873	35.423
Mot20-03	6.685	24.451	26.661
Mot20-05	2.731	16.452	19.739
Combined	9.386	22.252	**24.933**

TABLE III: comparing HOTA, baseline yolov5x6 model with the fine-tuned yolov5x6 model from experiments 2 and 3 on MOT-20 sequences.

Video Sequences	Experiment 1 HOTA	Experiment 2 HOTA	Experiment 3 HOTA
Mot20-01	50.392	41.098	42.492
Mot20-02	46.729	38.197	38.660
Mot20-03	44.431	21.450	23.890
Mot20-05	38.751	15.105	19.787
Combined	**41.95**	22.343	**24.988**

TABLE IV: comparing HOTA, baseline crowdhuman model with the fine-tuned crowdhuman model from experiments 2 and 3 on MOT-20 sequences.

VII. CONCLUSIONS

In conclusion, while working on the project we have gone through a considerable variety of deep learning-based multi-object tracking approaches like YOLOv5 + DeepSORT, YOLOv5 + StrongSORT, FairMot... and some state-of-the-art tracking approaches like TransCenter.

In addition, in TRDP-II, during the fine-tuning phase of the detector, initially with 8990 images and further with 14384 images we could observe that generally with an increase in the number of training images the accuracy of the algorithm increases which demonstrates the potential of synthetic data and how it can be used as a replacement for real data on tasks such as pedestrian detection, re-identification, and tracking. Also, it can be observed that after fine-tuning the crowdhuman model, for MOT-20 the accuracy increases by about 2.5% from experiment 2 to experiment 3, whereas for MOT-16 it increases by about 1%. After fine-tuning the yolov5x6 model, for the MOT-20 dataset, it increases by about 2.5% and for MOT-16 the accuracy increases by about 1.5% from experiment 2 to experiment 3.

Furthermore, during the TRDP-II we have established a better and deeper understanding of the fundamentals of multi-object tracking. From the obtained results, we can say that the synthetic training performs favorably, indicating that MOTSynth can completely replace manually annotated datasets while increasing performance. Hence, we can conclude that we have accomplished our goal defined for the TRDP-II.

On the other hand, during the working sessions, we 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 the 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 satisfied with the knowledge gained during the whole research process.

VIII. FUTURE WORK

After reviewing all the concepts and methodologies explained in the report, it becomes clear that the MOTSynth dataset has huge potential. Using all the videos in the MOTSynth dataset for training will further improve the results. Also, we would like to study the behavior of the algorithm after training the detector with synthetic sequences containing fewer people and testing the impact on the MOT16 dataset.

In addition, since in this project we mainly focused on the detector, we could also try to fine-tune the REID part of the StrongSORT tracker.

On the other hand, after our research about MOT, we found that there is, what seems to be, a game-changer method called TransCenter based on Transformers. Therefore, It would be interesting to further explore the mathematical background behind it and try to apply them and compare them with the

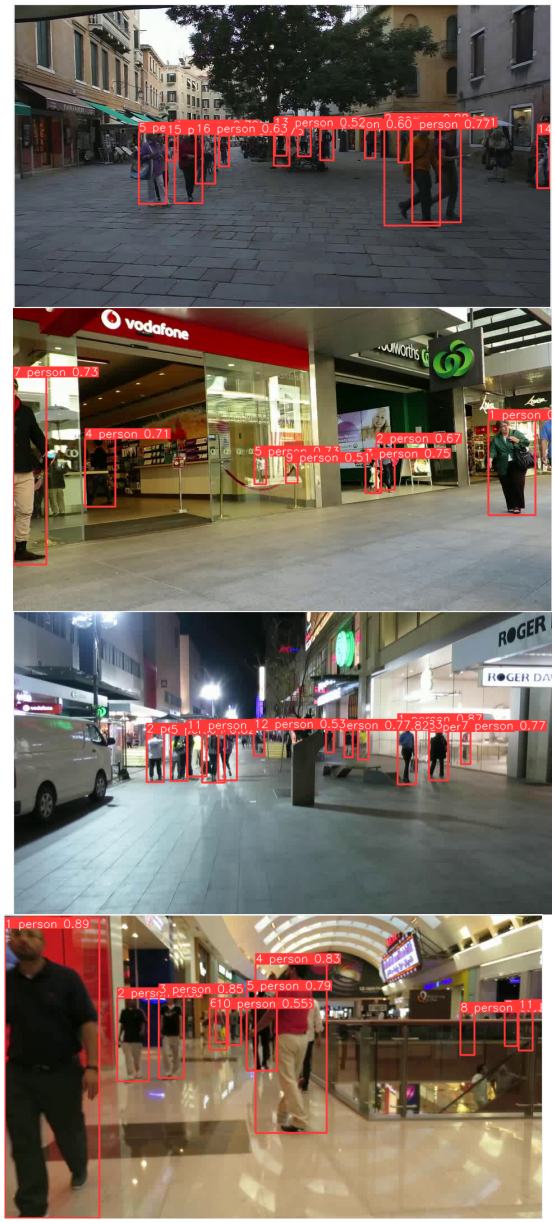

Fig. 7: results for the sequences MOT16-02, MOT16-09, MOT16-10, MOT16-11 (from top to bottom) using the fine-tuned crowdhuman model (from experiment-3).

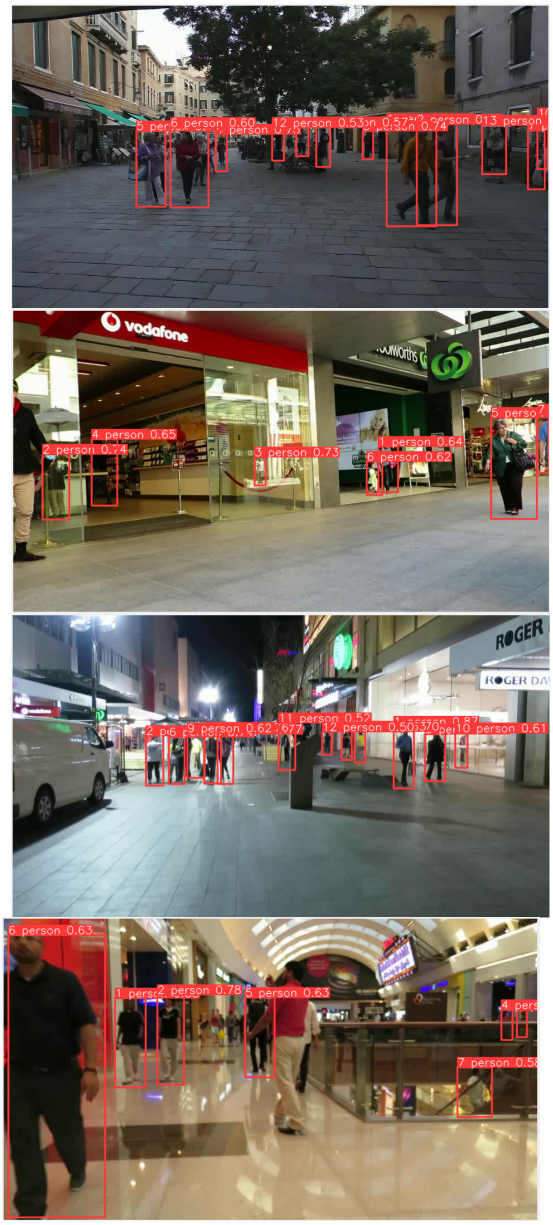

Fig. 8: results for the sequences MOT16-02, MOT16-09, MOT16-10, MOT16-11 (from top to bottom) using the fine-tuned yolov5x6 model.

previous results.

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] M. Broström, "Real-time multi-camera multi-object tracker using yolov5 and strongsort with osnet." https://github.com/mikel-brostrom/Yolov5_StrongSORT_OSNet, 2022.

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

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[15] C. Wu, W. Wen, T. Afzal, Y. Zhang, Y. Chen, and H. Li, "A compact DNN: approaching googlenet-level accuracy of classification and domain adaptation," *CoRR*, vol. abs/1703.04071, 2017.

[16] D. Kim, S. Park, D. Kang, and J. Paik, "Improved center and scale prediction-based pedestrian detection using convolutional block," in *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 418–419, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014.

[18] Z. Li, X. Tian, X. Liu, Y. Liu, and X. Shi, "A two-stage industrial defect detection framework based on improved-yolov5 and optimized-inception-resnetv2 models," *Applied Sciences*, vol. 12, no. 2, 2022.

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

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

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

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

[23] Y. Du, Y. Song, B. Yang, and Y. Zhao, "Strongsort: Make deepsort great again," 2022.

[24] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu, "A strong baseline and batch normalization neck for deep person re-identification," *CoRR*, vol. abs/1906.08332, 2019.

[25] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. J. Smola, "Resnest: Split-attention networks," *CoRR*, vol. abs/2004.08955, 2020.

[26] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *CoRR*, vol. abs/1609.01775, 2016.

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

[28] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Osep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?," *CoRR*, vol. abs/2108.09518, 2021.

[29] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *CoRR*, vol. abs/1605.06457, 2016.

[30] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *CoRR*, vol. abs/1805.00123, 2018.

[31] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, pp. 1–31, 02 2021.

Fig. 9: results for the sequences MOT20-01, MOT20-02, MOT20-03, MOT20-05 (from top to bottom) using the fine-tuned crowdhuman model(from experiment-3).

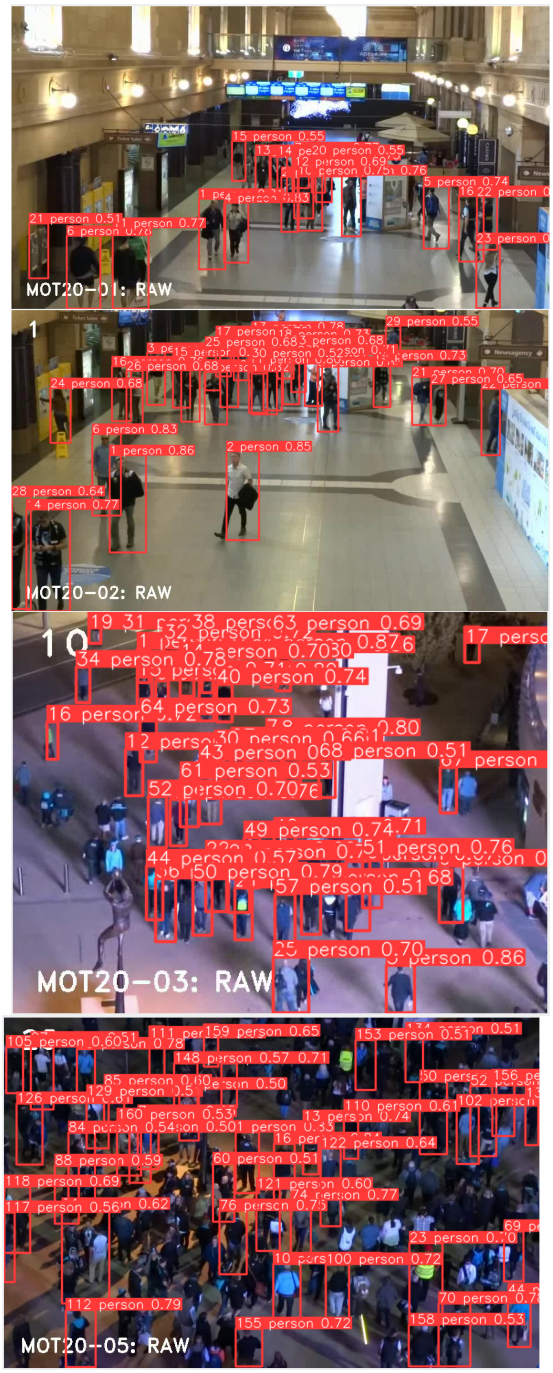

Fig. 10: results for the sequences MOT20-01, MOT20-02, MOT20-03, MOT20-05 (from top to bottom) using the fine-tuned yolov5x6 model.