Multiple Object Tracking: Combining YOLOv7 and deepSORT

Zhenhao Zhao
Dept. of Biomedical Engineering
The George Washington University
Washington, DC, 20052, USA
zzhao98@gwu.edu

***Abstract*—In recent years, computer vision (CV) has gradually attracted more and more attention in the industry. The CV algorithm greatly reduces labor costs in areas such as autonomous driving and express logistics, and improves the overall operating efficiency of the system. multiple Objects Tracking (MOT) algorithm is that without knowing the number of objects in advance, multiple objects such as pedestrians, cars, and animals in the video are detected and given identifications (IDs) for trajectory tracking. Different objects have different IDs, so as to realize subsequent trajectory prediction, precise search and other work. In this research, I will focus on MOT tasks, I will propose a new MOT algorithm by combining state-of-art object detection model, YOLOv7 to the existing tracking algorithm, deepSORT. I also deployed this MOT algorithm on a drone video, and the result show that my algorithm achieves good performance.**

***Index Terms*—Computer Vision (CV), Artificial intelligence (AI), multiple objects tracking (MOT)**

I. Introduction

In recent years, with the deepening of the research and application of deep learning and artificial intelligence, computer vision technology has made great progress in both the principle level and the application level. Unmanned Aerial Vehicle (UAS) technology, autonomous driving technology, and medical image processing technology based on computer vision, have disruptive innovations and changes in their respective fields. With the vigorous development of computer vision technology, CV tasks are gradually developing towards diversification. From traditional tasks such as classification and segmentation, more detailed and advanced tasks such as multiple object detection, multiple trajectory prediction and behavior recognition detection are gradually derived.

multiple Objects Tracking (MOT) algorithm is that without knowing the number of objects in advance, multiple objects such as pedestrians, cars, and animals in the video are detected and given identifications (IDs) for trajectory tracking. Different objects have different IDs, so as to realize subsequent trajectory prediction, precise search and other work. Now the MOT task is highly valued in the industry, because the algorithm has great potential in key fields such as automatic driving, intelligent transportation, and drone navigation. If this algorithm is deployed on a large scale, it will greatly reduce the overall operating costs of society and make people's lives more convenient and efficient.

In this research, I will focus on MOT tasks, and achieve better performance than before by combining two industry-recognized state of art algorithms - YOLOv7 [1] and Simple Online and Realtime Tracking with a Deep Association Metric (deepSORT) [2]. As the detection based MOT algorithm, it has a very important position in the industry because of its robustness and high precision. But in its algorithm proposed in 2017, the model part for detection uses the Faster RCNN mode [3]. Compared with YOLOv7, which was proposed in August 2022, the Faster RCNN model lags behind in both accuracy and inference time. Combining the two algorithms will help improve the overall performance of the MOT algorithm.

II. Related Work

A. Multiple Object Detection

In the computer vision field, much attention has been spent developing computer vision based object detectors. The Histogram of Oriented Gradients (HOG) [4] and Deformable Part-based Model (DPM) [5] were the peak of traditional detection based models. In 2014, a category of models called two-stage detectors was pioneered by R-CNN [6]. Incremental performance improvements were made by other models such as Fast R-CNN [7] and Faster R-CNN [3]. This group of models features a distinct region proposal module and classification module, which remained a bottleneck for real-time applications. In 2015, Redmon et al. proposed a single-stage detector, YOLO [8] which greatly sped up detection time, but sacrificed in accuracy. Then RetinaNet introduced a new loss function Focal Loss which helped the model achieve high inference speed while outperforming two-stage detectors in terms of accuracy. Single stage detectors are still amongst the state-of-the-art, including the YOLO family of models. In this work we used YOLOv7 [1] as our backbone detector of our MOT algorithm. YOLOv7 surpasses all known object detectors in bothspeed and accuracy in the range from 5 FPS to 160 FPSand has the highest accuracy 56.8% AP among all knownreal-time object detectors with 30 FPS or higher on GPU V100.

B. Multiple Object Tracking

In recent years, the development of the MOT algorithm has gradually improved, and data sets such as MOT Challendge [9], KITTI [10], and UA-DETRAC [11] have been born.

The emergence of these data sets has greatly stimulated the research of various scholars in different directions. Evaluation indicators such as Multiple Object Tracking Accuracy (MOTA) [12] have also been gradually improved. The early classic methods include Meanshift [13] and particle filter [14], but the overall accuracy is low, and they are mainly single-object tracking.

The current MOT research is mainly based on three frameworks:

1) MOT based on the combination of detection and tracking: the neural network directly outputs the object detection results and tracking results. Representative methods are JDE [15], CenterTrack [16], ChainedTracker [17], etc.

2) MOT based on attention mechanism: With the popular application of attention mechanism such as Transformer [18] in computer vision, some researchers have recently proposed a multiple object tracking framework based on attention mechanism. At present, there are mainly TransTrack [19] and TrackFormer [20], both works apply Transformer to MOT.

3) MOT based on detection: Firstly performs object detection on each frame of the video sequence, and then crops the object according to the bounding box to obtain all the objects in the image. After that, it is transformed into the objects association problem between the two frames to find the track. The similarity matrix is constructed through Intersection Over Union (IOU) and the feature vectors. This association problem can be solved by Hungarian algorithm, greedy algorithm and other methods. Simple Online and Realtime Tracking (SORT) [21] and its upgraded version: DeepSORT algorithm are both classic representatives of this idea.

In this study, I will propose a new MOT algorithm based on the third frameworks. Specifically, A state-of-art detector will be integrated in the original DeepSORT [2] to achieve better performance.

III. Methods

A. Dataset

We trained the object detection models on the VisDrone2019-Det dataset [22], which is a public dataset collected by the AISKYEYE team at the Lab of Machine Learning and Data Mining, Tianjin University, China. The dataset consists of 7,019 static images captured by various drone platforms in 14 different cities throughout China. The images are taken from a variety of altitudes and camera angles, covering a broad distribution of locations from urban to rural with a variety of object densities. This makes the model trained on this dataset robust and very suitable for this study.

B. Detector

The general structure of YOLOv7 is similar to the former YOLO model. It is composed of three key structures: the head, neck, and backbone. The input of YOLOv7 is an image or one frame of the video and the output is the following location vector and a classification vector for every pixels in the feature maps:

$$(x, y, w, h, o) \tag{1}$$

Here, x, y is the center 2D coordinates of the object bounding box and w, h is the width and height respectively. The o is if the object is the foreground information or the background information.

The backbone is a convolutional network to extract and process key features from the images. The neck will combine and analyze the features from the backbones, with fully connected layers, to make predictions on probabilities and spatial information for the bounding box. The head is the final output layer of the network that generates anchor boxes for feature maps and outputs final output vectors with class probabilities and bounding boxes of detected objects. [1] The detailed structure of YOLOv7 has been shown in the Fig. 1.

After post-processing such as non-maximum suppression, the final output is a python list, the length of the list is equal to the objects number:

$$[new\_bbox, cls\_conf, cls\_ids] \tag{2}$$

Here, new_bbox is a a vector like (x, y, w, h), is the location information of the object. cls_conf is the confidence of the classification and cls_ids is the classification identification.

C. Tracker

The tracker we used is deepSORT algorithm. It is the upgraded version of the SORT algorithm.

1) SORT Algorithm: The SORT algorithm uses the Kalman filter algorithm to predict the state of the detection frame in the next frame, and matches the state with the detection result of the next frame to realize multiple object tracking.

Kalman prediction: In SORT, the Kalman filter is used to predict the motion of the detection frame, then the following four states are required to describe a detection frame, namely: (1) The abscissa of the center of the detection frame u (2) The vertical coordinate of the center of the detection frame v (3) The size of the detection frame (called scale or area in the original paper) s (4) aspect ratio.r The above four states can describe the basic information of a detection frame, but cannot fully describe the motion state information of a state, so the above-mentioned state change information (which can be regarded as the speed of change) needs to be introduced to describe the motion state information. (1) The change speed of the abscissa of the center of the detection frame $\hat{u}$ (2) The change speed of the ordinate of the center of the detection frame $\hat{v}$ (3) The change speed of the size of the detection frame (called scale or area in the paper) $\hat{s}$ and (4) The change speed of aspect ratio $\hat{r}$. The whole state of each target is modelled as:

$$x = [u, v, r, s, \hat{u}, \hat{v}, \hat{r}, \hat{s}] \tag{3}$$

We input such x to the Kalman filter framework [23] with a new detection by YOLOv7, and we can get the update of the object.

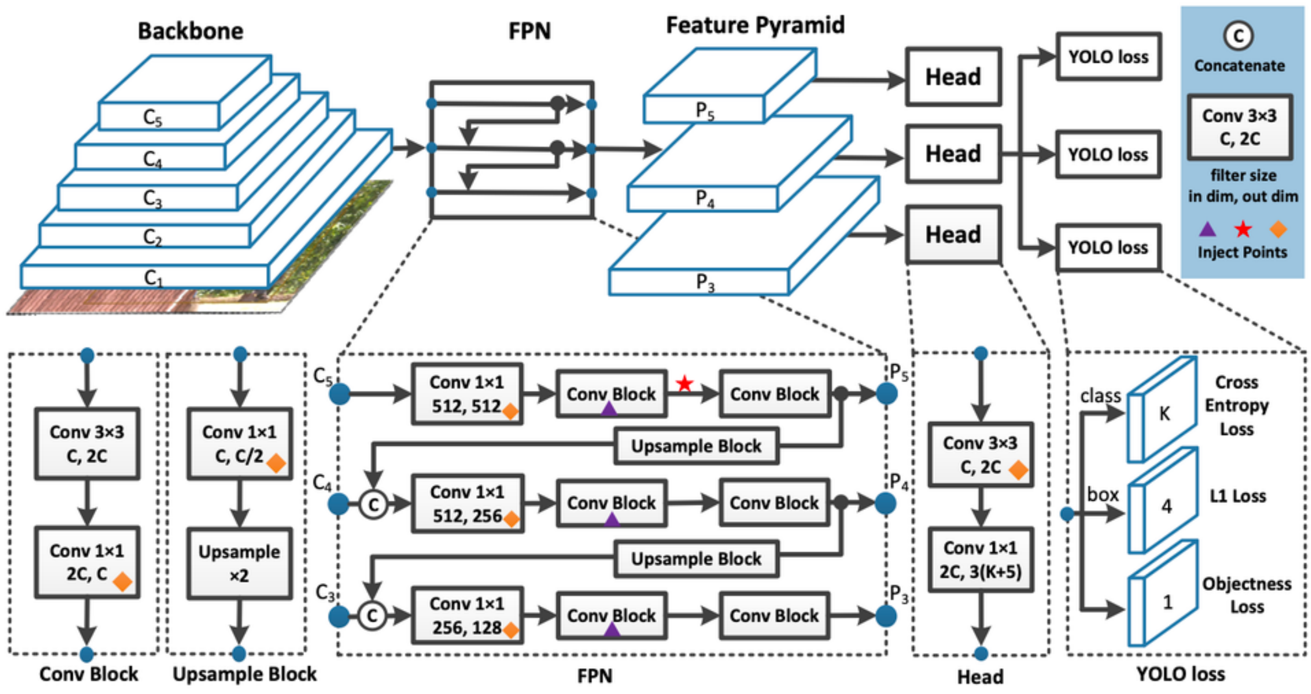


Fig. 1. The structure of the YOLOv7.

Data Association: SORT uses the Hungarian algorithm to match the bounding boxes predicted by kalman prediction and the bounding boxes output by YOLOv7 in real time. The Hungarian algorithm is a solution to the maximum matching problem of bipartite graphs. In addition, for each pair of matches, if the IOU is less than a certain threshold, such match will be rejected. For the unmatched detection, SORT will give it a new ID, for the unmatched track, the SORT will discard it. Fig. 2 show the general structure of SORT algorithm.

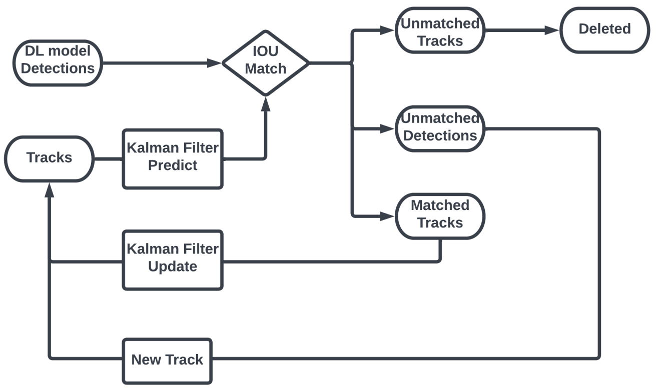


Fig. 2. The structure of the SORT.

2) DeepSORT Algorithm: In the SORT, once the object is occluded or not detected for other reasons, the state information predicted by the Kalman filter will not be able to match the detection result, and the tracking segment will end early. After the occlusion is over, the vehicle detection may be continued, so SORT can only assign a new ID number to the object, representing the beginning of a new tracking segment. Therefore, the disadvantage of SORT is that it is greatly affected by occlusion and other situations, and there will be a large number of ID switching.

Re-identification Network: The deepSORT overcome this issue by replacing the association metric with a more informed metric that combines motion and appearance information. In particular, they apply a convolutional neural network (CNN) that has been trained to discriminate objects with another large-scale relevant dataset. In the orignial paper, the dataset is a person re-identification dataset. Table. I show the structure of such CNN.

Deterministic States: In SORT, when the Kalman filter detects the result, it will update the result immediately, that is, to match the IOU with the detection frame. In Deepsort, we divide the detected results into deterministic states and non-deterministic states. Initially, they are all non-deterministic states. Only the matching of 3 consecutive frames (the matching here is similar to the direct IOU matching in SORT) can turn into a deterministic state, and the prediction result of turning into a deterministic state will perform Cascade Matching [2] on the root detection frame.

The whole structure of deepSORT is show in Fig 3.

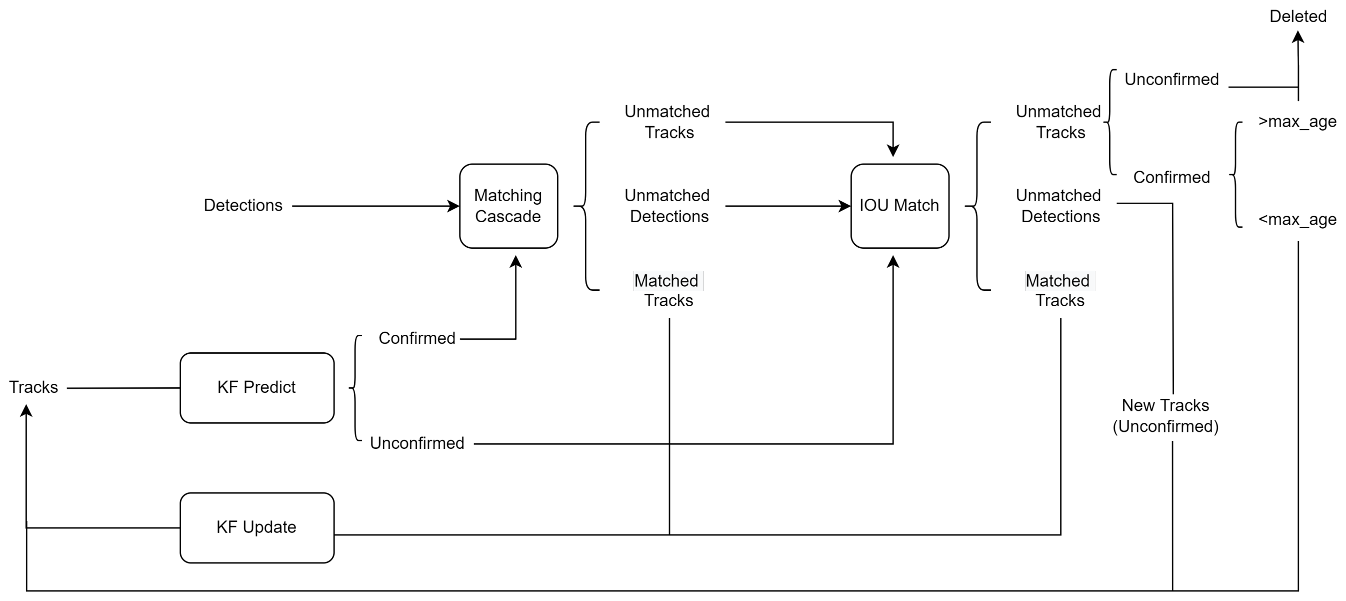


Fig. 3. The structure of the deepSORT

TABLE I
THE STRUCTURE OF THE RE-IDENTIFICATION CNN.

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

IV. RESULTS

A. Training and Evaluating the Detection Model

I present experimental results on the VisDrone2019-Det dataset. We use a workstation running Ubuntu 20.04 with an Intel Xeon W-3245 @ 4.60 GHz (32 cores), 128GB RAM, and a RTX 3070ti GPU (8GB) to train and evaluate the models. I also test the model in the real-world data collected during last summer semester.

The detection models are pre-trained on the COCO dataset [24]. The training and validation images were normalized and scaled, and data was augmented by horizontal image flipping. All training was done with an adam optimizer and optimal hyperparameters were selected through a hyperparameter grid search. The grid search was conducted with cross validation to determine the best learning rate, batch size, and patience over 5 epochs. The patience parameter is part of the scheduler which decreases the learning rate if the loss has plateaued over a number of epochs. The best performing hyperparameters were then selected to train the model over 20 epochs. Table II shows the result of the grid search result. Fig. 4 show the training process for the detection model. As the training progresses, the training and validation losses decrease, and model accuracy indicators increase. Mean Average Precision(mAP) is a metric used to evaluate object detection models. When the mAP is higher, the model should be more precise.

TABLE II
HYPERPARAMETER GRID SEARCH PERFORMANCE

Index	Learning rate	Batch size	Patience	Fold 0 mAP	Fold 1 mAP
1	0.001	4	1	0.0013	0.00075
2	0.001	4	3	0.0	0.0016
3	0.001	2	1	0.0023	0.0058
4	0.001	2	3	0.0	0.0016
5	0.00001	4	1	0.0168	0.0013
6	0.00001	4	3	0.0135	0.0013
7	0.00001	2	1	0.0138	0.0202
8	0.00001	2	3	0.0161	0.0203

The confusion matrix (Fig. 5) shows predicted classes versus the ground truth. The result is as expected since most of the predictions are along the main diagonal. Note that there are many more pedestrians and cars than other classes due to the high class imbalance of the dataset. The model did slightly misclassify other vehicle types as cars including vans, trucks, buses and motorcycles.

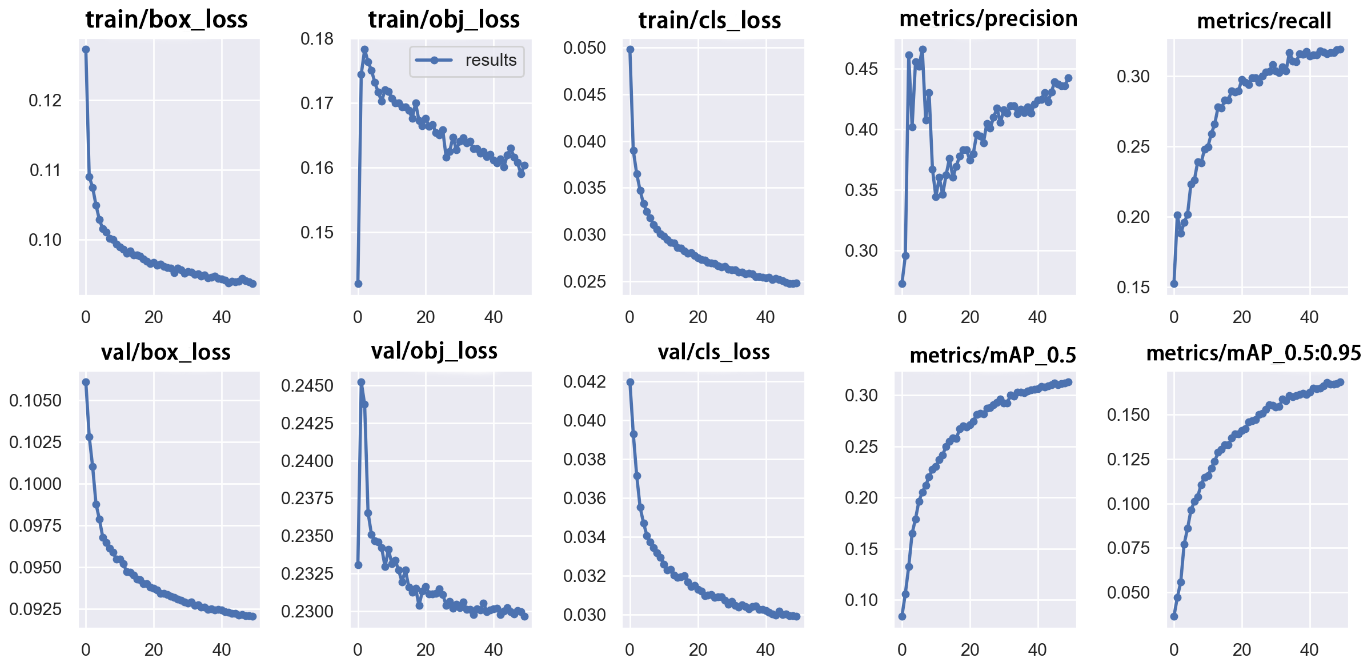
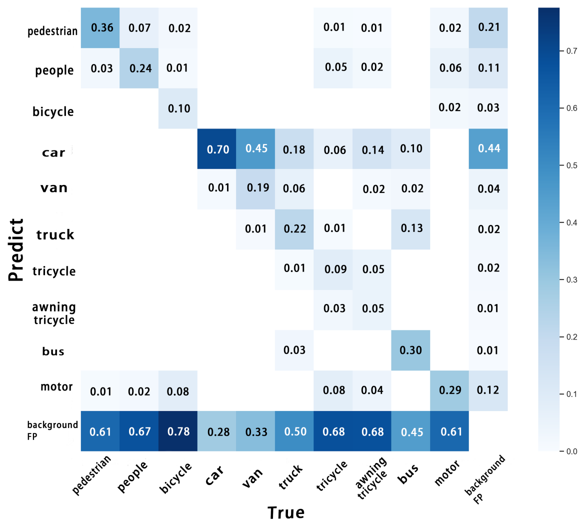


Fig. 4. The deep learning training plots for YOLO. The three columns on the left are the decreasing curves of various losses and the two columns on the right are the increasing curves of various accuracy indicators.



	pedestrian	people	bicycle	car	van	truck	tricycle	awning tricycle	bus	motor	background FP
pedestrian	0.36	0.07	0.02				0.01	0.01		0.02	0.21
people	0.03	0.24	0.01				0.05	0.02		0.06	0.11
bicycle			0.10							0.02	0.03
car				0.70	0.45	0.18	0.06	0.14	0.10		0.44
van				0.01	0.19	0.06		0.02	0.02		0.04
truck					0.01	0.22	0.01		0.13		0.02
tricycle						0.01	0.09	0.05			0.02
awning tricycle							0.03	0.05			0.01
bus						0.03			0.30		0.01
motor	0.01	0.02	0.08				0.08	0.04		0.29	0.12
background FP	0.61	0.67	0.78	0.28	0.33	0.50	0.68	0.68	0.45	0.61	

Fig. 5. Detection model confusion matrix.

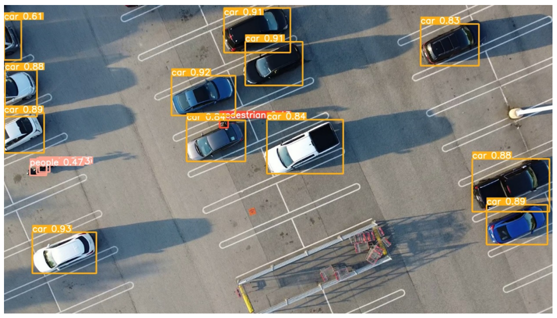


Fig. 6. YOLO model inference on one frame of the landing video.

Additionally, we tested the performance of the models on real landing videos taken on DJI Mini 2 drone (Fig. 6). The Mini 2 captured 4K videos, on which we ran the detection model post-flight.

B. Tracking Performance

The result of tracking is shown in the Fig. 7. The inference speed achieved 15 frames per seconds (FPS). This result shows that my algorithm has almost achieved the real-time tracking. And the accuracy of the algorithm also performed well, it can correctly track most of vehicles and pedestrians in the video.

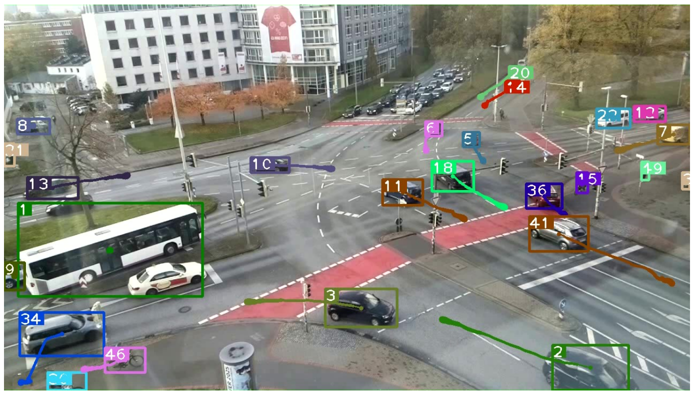


Fig. 7. Tracking performance of deepSORT algorithm.

V. Conclusion

In this research, I deployed YOLOv7 to the deepSORT algorithm, which achieved the desired effect and gained a deeper understanding of the tracking task. This tracking algorithm can be deployed in future scientific research and work to solve more problems. A limitation of this study is that due to limited time, there was not much comparison work with previous algorithms. We can perform more ablation experiments in the future research. [25]

REFERENCES

[1] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[2] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[7] R. Girshick, "Fast r-cnn," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. USA: IEEE Computer Society, 2015, p. 1440–1448. [Online]. Available: https://doi.org/10.1109/ICCV.2015.169

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] M. Weber, J. Xie, M. Collins, Y. Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, A. Ošep, L. Leal-Taixé, and L.-C. Chen, "Step: Segmenting and tracking every pixel," 2021. [Online]. Available: https://arxiv.org/abs/2102.11859

[10] 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.

[11] S. Lyu, M.-C. Chang, D. Du, W. Li, Y. Wei, M. Del Coco, P. Carcagnì, A. Schumann, B. Munjal, D.-H. Choi *et al.*, "Ua-detrac 2018: Report of avss2018 & iwt4s challenge on advanced traffic monitoring," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[12] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[13] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[14] Y. Boers and J. Driessen, "Particle filter based detection for tracking," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 6, 2001, pp. 4393–4397 vol.6.

[15] H.-W. Lin, V. M. Shivanna, H. C. Chang, and J.-I. Guo, "Real-time multiple pedestrian tracking with joint detection and embedding deep learning model for embedded systems," *IEEE Access*, vol. 10, pp. 51 458–51 471, 2022.

[16] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 474–490.

[17] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 145–161.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[19] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple object tracking with transformer," 2020. [Online]. Available: https://arxiv.org/abs/2012.15460

[20] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8844–8854.

[21] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[22] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," *CoRR*, vol. abs/1804.07437, 2018. [Online]. Available: http://arxiv.org/abs/1804.07437

[23] R. E. Kálmán, "New approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, 1960.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[25] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun, "Attention bottlenecks for multimodal fusion," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 14 200–14 213. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/76ba9f564ebbc35b1014ac498fafadd0-Paper.pdf