ONLINE SINGLE PERSON TRACKING FOR UNMANNED AERIAL VEHICLES: BENCHMARK AND NEW BASELINE

Zhihui Wang[1], *Zihao Liu*[1], *Dong Wang*[1*], *Shuang Wang*[2], *Yunwei Qi*[2], *Huchuan Lu*[1]

[1]School of Information and Communication Engineering, Dalian University of Technology
[2]Alibaba Group
Corresponding author: Dong Wang[*], wdice@dlut.edu.cn

ABSTRACT

Online tracking a specific person from a low-altitude unmanned aerial vehicle (UAV) is a very interesting and challenging problem to be solved. However, there exists no large-scale aerial video dataset regarding this online single person tracking (OSPT) task. To promote the study of the OSPT problem in UAV, we first construct a new benchmark dataset including 100 fully annotated aerial videos with nearly 130K frames and 11 challenging factors. Second, we evaluate several state-of-the-art online trackers with real-time performance using our dataset, considering the potential applications in the UAV platform. In addition, with respect to the OSPT problem, we attempt to design a new baseline method with the combination of tracking, detection and re-identification and conduct detailed analysis of different components. This method achieves much better performance than the existing online trackers, which will serve as a new baseline for our benchmark.

Index Terms— Object Tracking, UAV, Benchmark

1. INTRODUCTION

Surveillance over unmanned aerial vehicles (UAV) has drawn increasing attention with the rapid development of low-cost commercial UAVs. For a low-altitude UAV, accurately tracking a specific person will guide the drone to move and zoom automatically, resulting in a high-quality image observation. It brings many important applications such as suspect trailing, user following, outdoor navigation and event photography. The studies of this online single person tracking (OSPT) problem are limited without a well-established dataset.

Early UAV aerial datasets (e.g., VIVID [1] and CLIF [2]) have many limitations due to their small sizes, low-quality sequences or low frame rates, and focus on the vehicle targets from a high-altitude UAV view. In [3], Robicquet *et al.* compile a UAV campus dataset, which records 19K targets (e.g., pedestrians, bicyclists, cars and buses) from different top-view scenes. Such top-view imagery is suitable to trajectory analysis rather than visual tracking since the appearance information is limited in this condition. Although the UAV123 [4] and TB70 [5] datasets have presented for tracking generic objects in UAV, their scales and challenges are limited for recent trackers with respect to our OSPT task.

Fig. 1. Representative frames for our UAVP100 dataset. The ground truth of the tracked person is annotated by the red bounding box in each frame.

In this work, we construct a large-scale dataset for the OSPT task in a low-altitude UAV view. Representative frames are illustrated in Figure 1, from which we can see that our dataset contains a high diversity of scenes (roads, buildings, campuses, benches, squares, parks and fairgrounds), person appearances (e.g., many individuals, varied clothes and different activities), and challenging factors (see attribute analysis later). Our contributions can be summarized as three-fold.

First, we construct a fully annotated high-resolution dataset for the OSPT problem in a low-altitude UAV view and statistically analyze its differences compared with several related datasets. This dataset, named as UAVP100, consists of 100 aerial video sequences with nearly 130K frames. We note that our UAVP100 is a large-scale benchmark with respect to the OSPT problem in UAV.

Second, we evaluate 20 state-of-the-art online trackers with real-time performance and report their tracking results using two popular metrics and with various attributes. This evaluation facilitates the researchers to understand the tracking performance of existing real-time online trackers.

Third, we design an OSPT algorithm by integrating the online tracking, person detection and person re-identification into a unified framework. This method achieves significant improvement in comparison with state-of-the-art online trackers, which can be treated as a new baseline method.

2. OUR UAVP100 BENCHMARK

2.1. Dataset Construction

We construct a large-scale UAVP100 dataset for online tracking the category of persons in UAV whose altitudes varying between 5-30 meters. This dataset consists of 100 RGB video sequences captured using DJI Phantom 4, Inspire 2 and Spark drones, which includes totally **128913** frames with the 1080P (1920×1080) resolution. The frame rates of all videos are 30 frame per second (fps). We manually annotated the ground truth bounding boxes every frame, resulting in 125177 annotations as the targets disappear sometimes.

Table 1. Frame comparison of different datasets.

Dataset	UAVP100	UAV123-P	OTB100-P	UAV123	OTB100
#Video	100	55	36	123	100
#Min Frame	306	199	81	109	71
#Max Frame	2210	1783	1698	3085	3872
#Mean Frame	1291	1016	452	914	594
#Total Frame	128913	55861	16258	112467	58260

We show representative frames of UAVP100 in Figure 1, and then provide frame statistics in Table 1 (in comparison with UAV123 [4] and OTB100 [6]). The OTB100 dataset is one of the most popular benchmarks to evaluate online generic object tracking methods. The UAV123 dataset is most relevant to our dataset, which is designed for online single object tracking in UAV. The characteristics of our dataset will be better presented by comparing it with the two datasets above. We also create two subsets with respect to person tracking for further comparisons. For UAV123, we collect all sequences tracking persons to form a subset (named UAV123-P) to emphasize the OSPT task. In the same way, we collect an OTB100-P dataset from OTB100. Both total and average frame numbers in our UAVP100 dataset are significantly larger than in UAV123-P and OTB100-P and also competitive even compared with the original UAV123 and OTB100 ones.

2.2. Attribute Analysis

Evaluation with different attributes will facilitate researchers' understanding the advantages and limitations of a given tracker in dealing with different challenges (such as occlusion, background clutter, fast motion and so on). Motivated by the OTB100 [6] and UAV123 [4] datasets, the annotated sequences in our UAVP100 dataset are categorized into 11 attributes, which are defined as follows. (1) **Aspect Ratio Change (ARC)**: The quotient between the bounding box aspect ratio in the first frame and at least one subsequent frame is out of range [0.5,2]; (2) **Background Clutter (BC)**: The tracked object and its surrounding background have similar appearance; (3) **Fast Motion (FM)**: The center location difference of the tracked object in two consecutive frames is larger than 20 pixels; (4) **Full Occlusion (FOC)**: The tracked object is fully occluded; (5) **Illumination Variation (IV)**: The target region undergoes significant lighting changes; (6) **Long-term Disappearance (LD)**: The tracked object disappears in at least 60 consecutive frames (2 seconds) due to full occlusion or out-of-view; (7) **Out-of-View (OV)**: Most portion of the tracked object leaves the view; (8) **Partial Occlusion (POC)**: The tracked object is partially occluded; (9) **Similar Object (SOB)**: There exists objects (persons) of similar appearance near the tracked object; (10) **Scale Variation (SV)**: The ratio of the bounding box in the first frame and at least one subsequent frame is out of range $[0.5, 2]$; (11) **Viewpoint Change (VC)**: The viewpoint change affects the appearance of the tracked object significantly due to the drastic motion from either camera or object. Figure 2 illustrates the attribute analysis in our UAVP100 and the compared datasets. We can see that the attributes in our dataset cover more sequences, which also means each sequence includes more challenging factors.

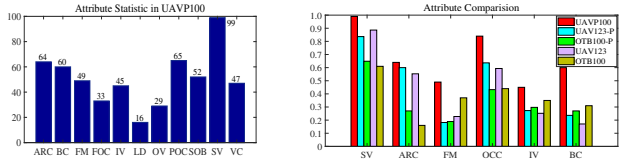


Fig. 2. Attribute analysis. Left: the sequence number of each attribute in UAVP100; Right: the ratios of representative attributes in different datasets.

2.3. Evaluation Protocol

We follow the one-pass evaluation protocol of OTB100 [6] and UAV123 [4], where different trackers are compared using both precision and success plots. The precision plot shows the percentage of frames whose center location error is smaller than a certain pixel; while the success plot illustrates the percentage of successfully tracked frames whose overlap is larger than a given threshold. Besides, different trackers are ranked based on the precision score at the threshold of 20 pixels for the precision plot and the area under curve (AUC) value for the success plot.

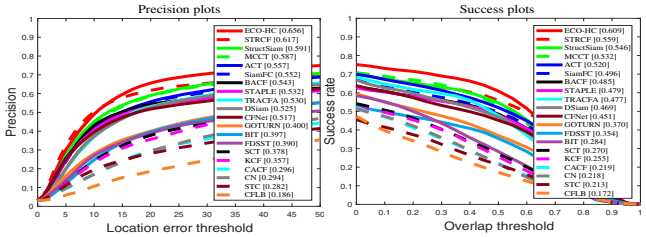
3. EVALUATION ON REAL-TIME TRACKERS

3.1. Compared Trackers

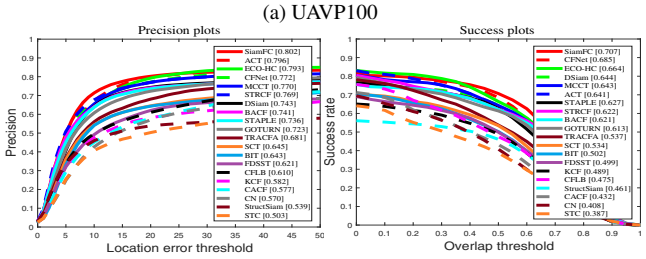
To consider the potential applications in UAV, we focus on evaluating state-of-the-art online trackers with real-time performance (more than 30*fps*). These trackers include C-N [7], STC [8], CFLB [9], KCF [10], BIT [11], SCT [12], STAPLE [13], SiamFC [14], GOTURN [15], FDSST [16], CFNet [17], ECO-HC [18], DSiam [19], CACF [20], BACF [21], STRCF [22], TRACA [23], MCCT [24], ACT [25], and StructSiam [26]. All above-mentioned twenty trackers are evaluated using our UAVP100 benchmark.

3.2. Experimental Results

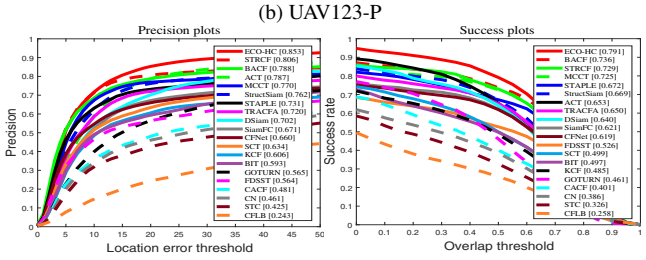
The above-mentioned trackers are tested on the PC platform with an Intel i7 3.4 GHz CPU with 32G memory and a Nvidia GTX 1080 GPU with 8G memory. We report the overall accuracies and average speeds in Figure 3 (a), the basic observations from which are as follows. First, the trackers with very fast speeds have not achieved satisfactory tracking accuracies (e.g., CN, STC and CFLB) since they merely exploit the basic correlation filter model with one single low-level hand-crafted feature (such as gray feature in STC, color feature in CN). Second, the Siamese-based tracking algorithms (SiamFC, StructSiam) achieve good results with real-time performance in GPU, where the lightweight deep networks are exploited to extract deep visual features taking a trade-off between robustness and efficiency. Third, the top-ranked trackers in CPU (ECO-HC, STRCF and MCCT) are designed based on the improved correlation filter models, the combination of color and texture features, or both. In addition, we report these trackers' performance in UAV123-P and OTB100-P datasets for comparisons (see Figure 3 (b-c)). Compared with Figure 3 (a), we can see that the accuracies of trackers in our UAVP100 dataset are much lower than those in UAV123-P and OTB100-P. This comparison further indicates that our dataset poses more challenges for evaluating a robust and efficient tracker, especially in a low-altitude UAV view.



(a) UAVP100



(b) UAV123-P



(c) OTB100-P

Fig. 3. Comparisons of different online trackers in the UAVP100, UAV123-P and OTB100-P datasets.

4. NEW BASELINE

4.1. Tracking Framework

In this work, our goal is to study the online single person tracking (OSPT) problem in UAV. It should be better to simultaneously consider the merit of online trackers and the information of person objects for designing a robust tracker. Thus, we attempt to design a tracking framework by combining the online tracker, person detection and person re-identification, the overall flowchart of which is illustrated in Figure 4. There exist three basic modules in our framework, including **Tracker**, **Verifier** and **Detector-ReID**. For a given frame, the **Tracker** module (with an online tracker) is exploited to determine the object location if the tracker's output is reliable (judged by **Verifier**). However, if the **Verifier** module treats the tracker being unreliable we will resort to the **Detector-ReID** module to re-locate the tracked person.

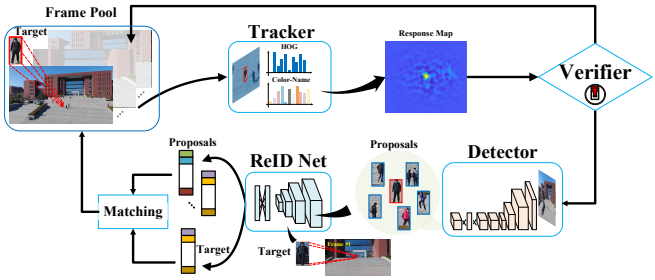


Fig. 4. The overall flowchart of our new baseline method.

Tracker: We adopt the ECO-HC [18] method as our **Tracker** module due to its high accuracy as well as fast speed. In addition, the ECO-HC tracker could generate a confidence map for the target in each frame, which facilitates the localization of the object and verification of the trackers' reliability.

Verifier: We design a simple but effective verification rule that takes the confidence map generated by the tracker as the input. For a normalized confidence map M, its peak value (i.e., $\max(M)$) should be large enough since it indicates the candidate score being most likely to the tracked object. In addition, this peak value should be much stronger compared with the values of other positions, which can be effectively measured by the Peak-to-Sidelobe Ratio (PSR) [27]. The PSR of the confidence map M is defined as

$$PSR(M) = (\max(M) - \mu_{\Phi}(M))/\sigma_{\Phi}(M), \quad (1)$$

where Φ is the sidelobe area around the peak region with 15% of the confidence map area. $\mu_{\Phi}(M)$ and $\sigma_{\Phi}(M)$ denote the average value and standard deviation of M except area Φ, respectively. The output of our **Verifier** can be designed as

$$\textbf{Verifier}(M) = \begin{cases} 1, \ \max(M) > tr_1, PSR(M) > tr_2 \\ 0, \ otherwise \end{cases} \quad (2)$$

to consider both absolute and relative strengths of the peak value. The **Verifier** outputting 1 means that the tracker is

reliable and is used to locate the tracked object in the current frame. Otherwise, it means that the tracker is unreliable and we will resort the person detection and re-identification scheme to re-initialize the location of the target. In practice, we choose $tr_1 = 0.6$ and $tr_2 = 13$ based on our empirical observations. In addition, we merely use the **Verifier** module every 10 frames to balance the accuracy and speed.

Detector-ReID: There exist no large-scale dataset for training person detection and re-identification models with respect to our OSPT task, which brings difficulties for designing our **Detector-ReID** module. Since the person appearances in the low-altitude UAV view have some similar characteristics with those in normal surveillance scenes, we believe the popular detection and re-identification algorithms could help us solve the OSPT problem in UAV to some degree. This work adopts the YOLOv2 [28] detector trained on ImageNet and the DGD ReID method [29] trained with several ReID datasets, to implement our baseline within the framework in Figure 4.

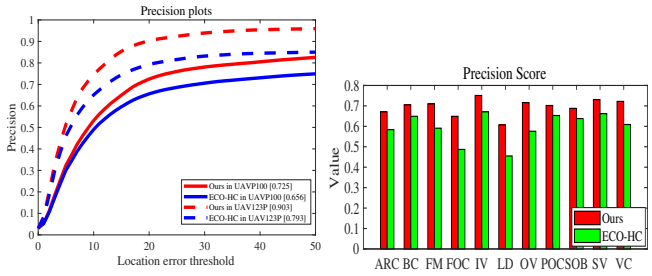
When the **Verifier** outputs 0, the **Detector-ReID** module is applied. First, the YOLOv2 detector generates a series of proposals regrading the persons in the current frame. Then, the DGD ReID network [29] extracts the deep features for both template (cropped in the first frame) and generated person proposals. Finally, the target's location will be re-initialized using the detection result with the smallest matching distance if this distance is smaller than a given threshold $tr_3 = 1.1$. Otherwise, we will attempt to conduct **Verifier** and **Detector-ReID** every frame until both **Tracker** and **Detector-ReID** modules are reliable.

We note that our baseline method (in Figure 4) is a general framework for the OSPT task in UAV, which makes the solution of our task benefit from any technological development of tracking, detection or re-identification.

4.2. Experimental Results

Quantitative Evaluation: In Figure 5 (a), we report the overall performance of the proposed new baseline algorithm (denoted as Ours) compared with the original ECO-HC method. Our new baseline tracker achieves a relative performance improvement of 10.5% in precision score. The generalization ability of our tracker is evaluated using UAV123-P, and the results show that it also achieves a significant improvement. In addition, the accuracies of compared methods in our UAVP100 are much lower than those in UAV123-P. Thus, we can conclude that our UAVP100 dataset is very challenging for the OSPT task.

Figure 5 (b) shows that the performance of our new baseline and ECO-HC methods on different attributes, in terms of precision score. We can see that our new baseline tracker consistently improves the tracking performance on all 11 attributes. Among them, significant improvements have achieved in handling Long-term Disappearance (LD), Full Occlusion (FOC) and Out-of-View (OV) challenges.



(a) (b)

Fig. 5. The overall performance of our new baseline.

Comparisons of Different Components: We test our baseline method with varied online trackers, pre-trained detectors and re-identification methods. First, a good online tracker facilitates improving the overall performance of the entire framework. We implement some variants with ECO-HC, SiamFC and FDSST, whose success scores are 0.725, 0.563 and 0.427. Second, we test our baseline method with some popular detectors including YOLOv2 [28], Faster R-CNN [30], RPN [31], SSD [31] and ACF [32] methods, achieving 0.725, 0.716, 0.692, 0.678 and 0.515 success scores. Thus, it is reasonable to choose YOLOv2 as the basic detector in our baseline method due to its efficiency. Third, we compare popular deep re-identification algorithms in our tracking framework. These methods include DGD [29], PartReID [33], SVDNet [34], DEP [35] and MTDNet [36]. Their corresponding success scores are 0.675, 0.651, 0.639, 0.634 and 0.634, respectively. Among them, the DGD method [29] works the best due to the powerful deep ReID features learned from multi-domain datasets. We also implement a baseline ReID method using the L2 distance with RGB color features, whose success score is only 0.609. This indicates that the study of person re-identification could facilitate the OSPT task in UAV.

5. CONCLUSIONS

This work constructs a large-scale dataset for online tracking persons in the view of low-altitude UAVs. Using this dataset, we evaluate 20 state-of-the-art real-time online trackers and report the detailed results in terms of both precision and success plots, which facilities the readers' better understanding of their potential applications in UAV. Finally, we design a baseline framework by combining online trackers, pre-trained detectors and re-identification methods. The results show that our tracker achieves much better performance, which will be acted as a new baseline for our benchmark.

Acknowledgement. This paper was supported in part by the Natural Science Foundation of China #61751212, #61872056, #61725202, and in part by the Fundamental Research Funds for the Central Universities under Grant #DUT18JC30. This work was also supported by Alibaba Group through Alibaba Innovative Research (AIR) program.

6. REFERENCES

[1] Robert Collins, Xuhui Zhou, and Seng Keat Teh, "An open source tracking testbed and evaluation web site," in *PETS Workshop*, 2005.

[2] Vladimir Reilly, Haroon Idrees, and Mubarak Shah, "Detection and tracking of large number of targets in wide area surveillance," in *ECCV*, 2010, pp. 186–199.

[3] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *ECCV*, 2016, pp. 549–565.

[4] Matthias Mueller, Neil Smith, and Bernard Ghanem, "A benchmark and simulator for UAV tracking," in *ECCV*, 2016, pp. 445–461.

[5] Siyi Li and Dit-Yan Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *AAAI*, 2017, pp. 445–461.

[6] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[7] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, 2014, pp. 1090–1097.

[8] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang, "Fast visual tracking via dense spatio-temporal context learning," in *ECCV*, 2014, pp. 127–141.

[9] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey, "Correlation filters with limited boundaries," in *CVPR*, 2015, pp. 4630–4638.

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

[11] Bolun Cai, Xiangmin Xu, Xiaofen Xing, Kui Jia, Jie Miao, and Dacheng Tao, "BIT: biologically inspired tracker," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1327–1339, 2016.

[12] Jongwon Choi, Hyung Jin Chang, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi, "Visual tracking using attention-modulated disintegration and integration," in *CVPR*, 2016, pp. 4321–4330.

[13] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401–1409.

[14] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCVW*, 2016, pp. 850–865.

[15] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 FPS with deep regression networks," in *ECCV*, 2016, pp. 749–765.

[16] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[17] Jack Valmadre, Luca Bertinetto, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017, pp. 5000–5008.

[18] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg, "ECO: efficient convolution operators for tracking," in *CVPR*, 2017, pp. 6931–6939.

[19] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017, pp. 1781–1789.

[20] Matthias Mueller, Neil Smith, and Bernard Ghanem, "Context-aware correlation filter tracking," in *CVPR*, 2017, pp. 1387–1395.

[21] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, 2017, pp. 1144–1152.

[22] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *CVPR*, 2018, pp. 4904–4913.

[23] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewang Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi, "Context-aware deep feature compression for high-speed visual tracking," in *CVPR*, 2018, pp. 479–488.

[24] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li, "Multi-cue correlation filters for robust visual tracking," in *CVPR*, 2018, pp. 4844–4853.

[25] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu, "Real-time 'actor-critic' tracking," in *ECCV*, 2018, pp. 328–345.

[26] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu, "Structured siamese network for real-time visual tracking," in *ECCV*, 2018, pp. 355–370.

[27] Yang Li, Jianke Zhu, and Steven C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *CVPR*, 2015, pp. 353–361.

[28] Joseph Redmon and Ali Farhadi, "YOLO9000: better, faster, stronger," in *CVPR*, 2017, pp. 6517–6525.

[29] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *CVPR*, 2016, pp. 1249–1258.

[30] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[31] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He, "Is faster R-CNN doing well for pedestrian detection?," in *ECCV*, 2016, pp. 443–457.

[32] Piotr Dollár, Ron Appel, Serge J. Belongie, and Pietro Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[33] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang, "Deeply-learned part-aligned representations for person re-identification," in *ICCV*, 2017, pp. 3239–3248.

[34] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang, "Svdnet for pedestrian retrieval," in *ICCV*, 2017, pp. 3820–3828.

[35] Zhedong Zheng, Liang Zheng, and Yi Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 1, pp. 1–20, 2018.

[36] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang, "A multi-task deep network for person re-identification," in *AAAI*, 2017, pp. 3988–3994.