Real-Time Traffic Monitoring with Occlusion Handling[*]

Mauro Fernández-Sanjurjo, Manuel Mucientes, and Víctor M. Brea

Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela, Santiago de Compostela, Spain
mauro.fernandez@usc.es, manuel.mucientes@usc.es, victor.brea@usc.es

Abstract. Traffic surveillance through vision systems is a highly demanded task. To solve it, it is necessary to combine detection and tracking in a way that meets the requirements of operating in real time while being robust against occlusions. This paper proposes a traffic monitoring system that meets these requirements. It is formed by a deep learning-based detector, tracking through a combination of Discriminative Correlation Filter and a Kalman Filter, and data association based on the Hungarian method. The viability of the system has been proved for roundabout input/output analysis with near 1,000 vehicles in real-life scenarios.

Keywords: Multiple Object Tracking · Traffic Monitoring · Roundabout Analysis

1 Introduction

Detection and tracking of vehicles in a video allows to estimate every vehicle trajectory while they remain in the scene. This has applications in a wide range of tasks: vehicle counting, accident detection, roundabout entry/exit analysis or assisted traffic surveillance. In a real-life scenario speed and robustness are a must, which translate to the requisites of real-time performance and occlusion handling.

In terms of the current tracking solutions we can distinguish two types: *low-level* and *high-level* trackers. The former exploits the visual information in the current frame to find the object of interest while the latter can use more complex information to estimate the new object position (probabilistic models, environment maps, etc.). Current *low-level* trackers [2,7,5] cannot handle total occlusions and do not provide a framework for multiple object tracking. In addition, the

[*] This research was partially funded by the Spanish Ministry of Economy and Competitiveness under grants TIN2017-84796-C2-1-R and RTI2018-097088-B-C32 (MICINN/FEDER), and the Galician Ministry of Education, Culture and Universities under grant ED431G/08. Mauro Fernández is supported by the Spanish Ministry of Economy and Competitiveness under grant BES-2015-071889. These grants are co-funded by the European Regional Development Fund (ERDF/FEDER program). We thank Aplygenia S.L. for their collaboration.

best current solutions require a high end GPU or do not operate in real time with multiple objects on CPU [7,20].

In recent years, the high-level tracking problem has been focused as a tracking-by-detection approach [1]. This framework considers the tracking task as a data association problem between detections and trackers over time. This assumes the existence of reliable detections in every frame of a video, something that in a real-life scenario is not a valid option as current state-of-the-art deep-learning based detectors operate above 75 ms per frame [17].

In this paper, we present a traffic monitoring system that performs multiple object detection and tracking in a video in real-time handling total occlusions. The system is composed of a deep-learning based detector, a low-level Discriminative Correlation Filter (DCF) based tracker, a high-level Kalman Filter based tracker and data association based on the Hungarian algorithm. The contributions of our proposal are:

- A traffic monitoring system that can process **more than 400 vehicles simultaneously** in videos with HD resolution in real-time.
- The system also **handles occlusions** by detecting the upcoming occlusion and searching the occluded vehicle in a zone called *ROI (Region-Of-Interest)* that is proportional to the error degree in the tracking process. We provide a metric for **on-line tracking failure detection** by estimating the distance between two independent tracking methods allowing us to update the system's tracking error accordingly.
- We extend our system for solving a **real-life traffic application**: roundabout I/O (Input/Output) with near 1,000 vehicles.

The rest of this paper is structured as follows. Section 2 gives an overview of closely related work. In Section 3 we explain the details of our approach. In Section 4 we discuss the implementation details of our system and introduce the traffic application developed. Finally, conclusions are given in Section 5.

2 Related Work

Traffic monitoring systems detect and track all the vehicles in a video sequence. This task presents two main challenges: to manage total occlusions and to operate in real-time with multiple vehicles.

The work in the field of object detection is mainly based on deep convolutional neural networks (ConvNets). One of the first works in this area was R-CNN [12] which uses a region proposal algorithm (such as selective search [23] or edge boxes [25]) and applies a classification network to each of them. Improving the previous approach, Fast-RCNN [11] introduces the regions in an intermediate stage of the network, thus, saving a lot of computing time. Finally, becoming the milestone in the object detection field, Faster-RCNN [22] introduces a region proposal algorithm based entirely on a neural network called the

Region Proposal Network (RPN). The RPN uses the information from intermediate layers of a standard classification network to provide different locations in which an object may appear.

To improve the performance of the proposal of regions in all possible scales, Lin et al. [18] replicate the RPN from Faster-RCNN in several layers of the network in which deeper feature maps are combined with shallower ones. The shallower the layer the smaller the object it will locate. This approach, called Feature Pyramid Network (FPN) obtains outstanding results as shown in the COCO detection challenge 2016 [19]. All these approaches present a high level of performance but, their main limitation is their computational cost, which makes them harder to use in applications that demand real-time performance.

In the last years, top trackers from the Visual Object Tracking (VOT) challenge [15] are based on two approaches: Discriminative Correlation Filters (DCF) based trackers, and deep-learning based trackers. On the one hand, DCF based trackers predict the target position training a correlation filter that can differentiate between the object of interest and the background [6,13,5]. On the other hand, deep-learning based trackers use ConvNets. SiamFC [2] is one of the first approaches of this kind. This tracker consists of two branches that apply an identical transformation —deep features extractor— to two inputs: the search image and the exemplar. Then, both representations are combined through cross-correlation, generating a score map that indicates the most probable position of the object.

Due to the increase in performance of deep learning detectors in recent years, the task of tracking is increasingly being seen as a data association problem, i.e. tracking-by-detection. In this approach, the primary concern is to assign detections to trackers over time. Some international challenges [1] have emerged to rank solutions to this problem, evaluating precision, robustness and speed among other performance metrics. In the past few years, complex solutions to this tracking approach that obtain outstanding results have appeared. Some of them focus on extending traditional high-level tracking approaches. As an example, Kim et al. [14] and Chen et al. [4] propose extensions to the classical multiple hypotheses tracking (MHT) [21]. The former introduces on-line appearance representations while the latter enhances the classical MHT by incorporating a detection model that includes detection-scene and detection-detection analysis.

All these approaches have demonstrated good performance in classic multiple object tracking metrics as commented before. Their fundamental limitation is the speed, as none of the work discussed in this section shows performance metrics above 2.6 Hz even without accounting for the detection time. Also, they assume the existence of detections in every frame of a video without taking into account high performance object detectors inference time.

Some work in the traffic monitoring field has been done in the recent years [8]. In [10], vehicle counting is performed employing an environment segmentation strategy. In [9] a tracking approach using background subtraction and Kalman filter tracking to tackle the data collection in roundabouts is proposed. These approaches usually run at real-time speed due to the use of background subtraction for detecting mobile objects. These object identification methods could represent a limitation in scenarios that present camera movement (on-board cameras), shadows, image artifacts, or objects that appear very close to each other since they usually are identified as only one by the background subtraction algorithm.

3 Video Traffic Monitoring

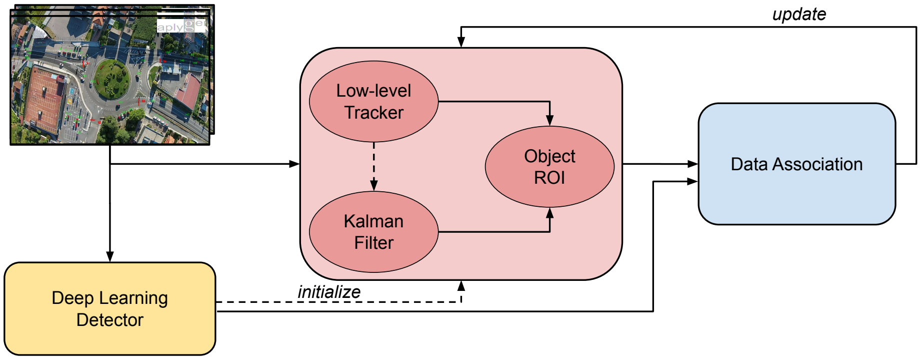


Fig. 1: Architecture of our traffic monitoring system. It is formed by three modules: detection (yellow), tracking (red) and data association (blue).

We propose a complete traffic monitoring system that combines tracking and detection and can operate as a baseline for multiple applications.

Our system is made up of three blocks (Figure 1): detection, tracking and data association. To detect vehicles in an image, we use a deep learning based detector. For tracking, we combine a DCF-based tracker with a Kalman-based one, which enables to calculate a failure detection metric to identify occluded vehicles. Finally, in the data association module, we assign each detection with its correspondent tracker through the Hungarian method [16,24] and perform an update of the trackers.

Algorithm 1 presents the main steps of the system. The inputs to the system at every time instant t are the new frame (Im_t) of the video, and the set of trackers in the previous time instant (Φ_{t-1}). First, the trackers positions in the new image (Im_t) are estimated. We start calculating the new position of the object with a DCF tracker (Algorithm 1, line 3 —Alg. 1:3—). Tracking based just on DCF trackers has two limitations: (i) we cannot handle occlusions (Figure 3); (ii) it does not provide a robust tracking failure detection (i.e. knowing when the tracking fails) as the PSR (Peak to Sidelobe Ratio) value [3], which measures the spread degree of the convolution operation of the correlation filter, is not a reliable measure. As shown in Figure 4, the PSR takes different threshold values for different videos and scenarios, which makes difficult to identify when a tracker is lost.

Algorithm 1: Traffic Monitoring System

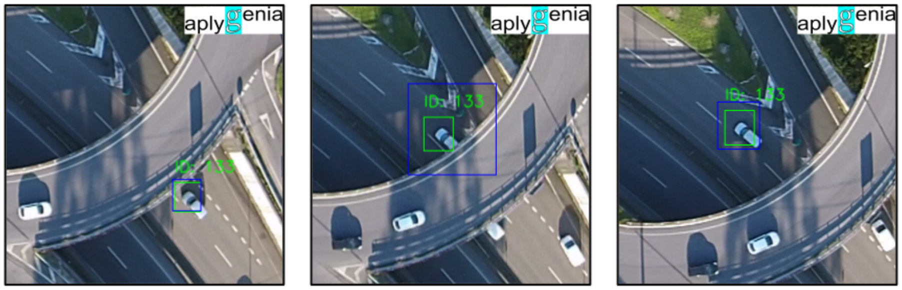
```
Require:
(a) $Im_t$: Image frame at current time $t$
(b) $\Phi_{t-1} = \{\varphi_{t-1}^1, \varphi_{t-1}^2, \ldots, \varphi_{t-1}^n\}$

1  Function Main($Im_t$, $\Phi_{t-1}$):
2      for $i$=1 to $n$ do
3          $dcf\_roi_t^i$ =DCF_Track ($\varphi_{t-1}^i$)
4          $kf\_roi_t^i$ = Kalman_Predict ($\varphi_{t-1}^i$)
5          $\overline{\varphi}_t^i \leftarrow < roi_t^i >$=Estimate_ROI ($dcf\_roi_t^i, kf\_roi_t^i$)
6      if $time\_elapsed > \tau$ then
7          $\Psi_t \leftarrow \{\psi_t^1, \psi_t^2, \ldots, \psi_t^m\}$ =ConvNet_Detect()
8          for $i$=1 to $n$ do
9              for $j$=1 to $m$ do
10                 $IOU_t^{i,j} = \frac{\overline{\varphi}_t^i \cap \psi_t^j}{\overline{\varphi}_t^i \cup \psi_t^j}$
11         $\{< \overline{\varphi}_t^\alpha, \psi_t^\beta >\}$ =Hungarian ($IOU_t$)
12         for every $\alpha, \beta$ in $\{< \overline{\varphi}_t^\alpha, \psi_t^\beta >\}$ do
13             update_tracker ($\overline{\varphi}_t^\alpha, \psi_t^\beta$)
14         for $i$=1 to $n$ do
15             if not_updated ($\overline{\varphi}_t^i$) then
16                 check_tracker_deletion ($\overline{\varphi}_t^i$)
17         for $j$=1 to $m$ do
18             if not_updated ($\psi_t^j$) then
19                 new_tracker ($\psi_t^j$)
20     $\Phi_t = \overline{\Phi}_t$
21     return ($\Phi_t$)
```

To provide a solution to both problems, we introduce a Kalman Filter (KF) tracker that, by modeling the movement of the object can handle occlusions and, in combination with the DCF tracker, can estimate the error in the tracking process. So, once the vehicle's new position is calculated by the DCF tracker, we estimate the position using the Kalman filter. We use a linear constant velocity model in the KF, so the state of each vehicle is modeled as:

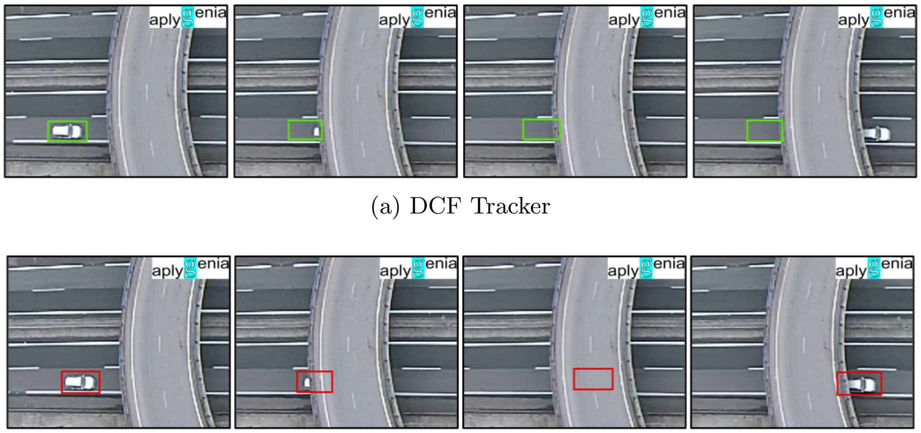
$$\mu := [x, y, v_x, v_y] \tag{1}$$

Here x and y are the position of the object, and v_x and v_y represent the linear velocity in both axes. We perform Kalman prediction in Alg. 1:4. With the bounding boxes proposed by both methods, we estimate the region of interest (ROI) in which the object might be located (Alg. 1:5). The larger the difference between the two trackers, the larger the ROI. Occlusions can be determined in cases where both predictors propose very different bounding boxes,



(a) (b) (c)

Fig. 2: Creation of a search ROI for occlusion handling. (a) Both tracking methods agree on the object position. (b) As the DCF fails to track the occluded object, the distance between both estimations increases and so it does the search ROI. Finally in (c), when the detector finds the vehicle at the other side of the road and the tracker recovers. Images courtesy of Aplygenia S.L.



(a) DCF Tracker

(b) DCF+KF Tracker

Fig. 3: (a) The low-level DCF tracker (in green) cannot recover the identity of the object once occluded as it only relies on appearance. (b) The combination of a DCF and a KF manages occlusions, as it also takes into account the object motion model. Images courtesy of Aplygenia S.L.

since the bounding boxes provided by DCF will remain static, while those from the Kalman filter will follow the previous movement pattern of the object (Figure 2).

Our system is robust enough so we do not need to call the detector in every frame. The aim of the detection component is twofold. First, it initializes every tracker or object of interest in the scene. Second, it refines the location and size

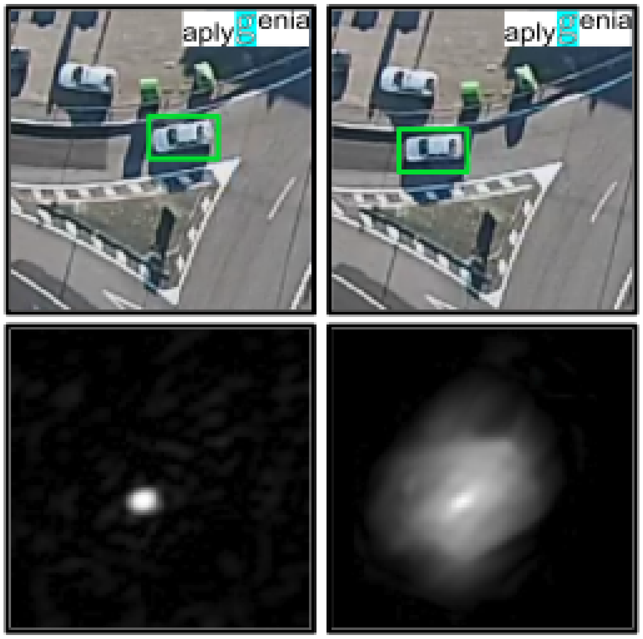


Fig. 4: PSR values are poor predictors of tracking failures for the DCF tracker. The image shows a case in which the object is being tracked successfully but the PSR value changes and shows a high degree of dispersion. The opposite case, a tracking failure not detected by the PSR values, is also frequent. Images courtesy of Aplygenia S.L.

of the bounding boxes of the trackers along their trajectories through the data association component (see Fig. 1), improving tracking performance metrics. If the time elapsed since the previous detection is greater than or equal to τ, detection is performed using a convolutional neural network (Alg. 1:6-7), which returns a set of detections Ψ_t. In practice, this is performed with a fully convolutional network called FPN [18], which uses feature maps information at different scales to locate from small to large objects, through a pyramidal architecture with lateral connections between them. The FPN provides high precision at a high computational cost, taking about 130 ms to perform a full detection in an HD image. If no detection is performed at current time t, tracking prediction alone ($\overline{\Phi}_t$) determines the current trackers state (Φ_t, Alg. 1:20).

The data association block aims to assign each detection to its corresponding tracker and to identify objects that enter or leave the scene. In so doing, we build up the cost matrix IOU_t (see Alg. 1:8-10), where every entry is the Intersection Over Union (IOU) between a tracker $\overline{\varphi}_t^i$ and a detection ψ_t^j. That association is solved by the Hungarian Method (Alg. 1:11). For every successful assignation ($< \varphi_t^{\alpha}, \psi_t^{\beta} >$), tracker φ_t^{α} is updated with detection ψ_t^{β} (Alg. 1:13). Finally, trackers not updated in the data association phase are candidates for being deleted, and detections not assigned are initialized as new trackers (Alg. 1:14-19).

4 Results

Tracking	
Frames processed by second	30 frames of 30
Total max. time with parallel computing	0.0121 sec (60 objects, 15 threads)
Max. number of objects in 0.03 sec	148 objects
Detection	
Frames processed by second	5 frames of 30
Average time per HD image	0.135 sec

Table 1: Computational times for the detection and tracking modules of the traffic monitoring system.

The proposed system (Figure 1) runs on a server with an Intel Xeon E52623v4 2.60 GHz CPU, 128 GB RAM and an Nvidia GP102GL 24GB [Tesla P40] as GPU. Table 1 shows the times of the two most computational expensive operations of our system: detection and tracking — computing times of other tasks are negligible. In a 30 fps video, we have 0.03 seconds per frame for the tracking task. Using 15 threads for parallelization, theoretically, the system is able to process up to 148 objects in the image while maintaining real-time performance, *i.e.* 30 fps. As mentioned before, detection is the slowest part of our system, taking an average 0.135 seconds in an HD image and 0.075 seconds in VGA resolution. These values are below the 0.2 threshold required by the system for the detection module, as we only perform detection 5 times every 30 frames.

4.1 Roundabout Monitoring

In this section, we analyze our complete system (Figure 1) for roundabout monitoring. The objective of the system is to identify the entry and the exit a vehicle takes, maintaining its identity while it remains in the roundabout. The final goal is to provide the I/O matrix R, in which every element ($R(i,j)$) represents the number of vehicles that joined the roundabout taking entry i and exit j. If a vehicle enters the roundabout and exits it with the same ID we count that as a

Tracking	
Frames processed by second	10 frames of 30
Total max. time with parallel computing	0.0121 sec (60 objects, 15 threads)
Max. number of objects in 0.1 sec	492 objects
Detection	
Frames processed by second	5 frames of 30
Average time per HD image	0.135 sec

Table 2: Computational times for the fast version of our traffic monitoring system.

tracking success. On the contrary, if the identity changes along the video, then we count that vehicle as a tracking failure.

For performing the metrics, we use a video dataset which consists of five videos of roundabouts recorded from an Unmanned Aerial Vehicle (UAV) at 30 fps with HD resolution [1]. The videos have different conditions that are challenging for traffic monitoring: shadows, total occlusions (two level roads), camera movement, etc. Figure 5 shows a snapshot of some of these videos [2].

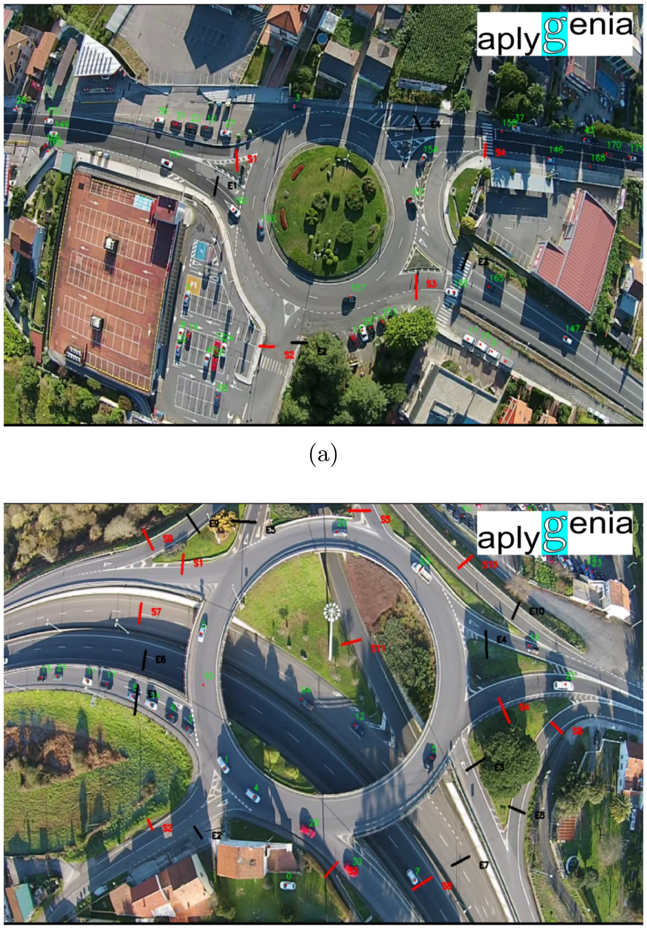
As explained before, the robustness of our system allows us to avoid calling the detector at every frame. This led us to develop a fast version that performs tracking in one of every 3 frames and detection in one of every 6 frames, without degrading the performance metrics for roundabout monitoring. Table 2 shows the times for this version fast version.

Video	**#occ**	**#vocc**	**Time (min:sec)**	**#vehicles**	**Success**
usc_vr_1	308	160	05:11	320	86,50%
usc_pl_1			11:12	138	88%
usc_rb_1			11:48	230	95%
usc_sx_1			09:26	255	91%
usc_ou_1	22	11	02:49	52	96%
Total	***330***	***171***	***40:26***	***995***	***91,30%***

Table 3: Results in the video dataset for roundabout monitoring. The columns are: video, number of occlusions (#*occ*), number of vehicles occluded (#*vocc*), duration of video, total number of vehicles (#*vehicles*) and success rate obtained by our tracking system.

[1] These videos where recorded by the company Apligenia S.L.

[2] A demonstration video can be downloaded from: `http://bit.ly/roundabout_sample_video`



(a)

(b)

Fig. 5: Example frames of some videos of the roundabout monitoring dataset. These videos are recorded from an UAV flying over a roundabout. Images courtesy of Aplygenia S.L., their distribution is restricted.

Table 3 shows the results obtained from processing the I/O matrix of five videos with 995 vehicles in total. We have used the fast version of our traffic monitoring system to highlight the robustness of the proposal even when processing just 10 of each 30 frames. Theoretically, the system can track up to 492 objects, although in these videos the maximum number of concurrent objects was 60. An average success rate of 91% is obtained. Results also show our system's ability to handle occlusions as two of the videos are scenarios with a high rate of total occlusions: in one of them the 50% of the vehicles are totally occluded nearly twice on average.

5 Conclusions

We have presented a traffic monitoring system that combines a convolutional neural network detection, DCF and Kalman trackers, and a Hungarian data association. The system is able to track hundreds of objects in real-time while being robust to occlusions. The combination of the DCF and Kalman filters allows to estimate the error of each tracker, thus increasing the robustness and reliability of the system. We have applied the traffic monitoring system to the problem of roundabout monitoring. Our system achieves a 91% success rate for the I/O matrix, even in cases with high occlusion rates, shadows and movement of the UAV onboard camera.

References

1. MOTChallenge the multiple object tracking benchmark. `https://motchallenge.net/`, accessed: 2018-12-18
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision Workshops (2016)
3. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010)
4. Chen, J., Sheng, H., Zhang, Y., Xiong, Z.: Enhancing detection model for multiple hypothesis tracking. In: IEEE Conference on Computer Vision and Pattern Recognition Workshop (2017)
5. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
6. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(8), 1561–1575 (2017)
7. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision (ECCV) (2016)
8. Datondji, S.R.E., Dupuis, Y., Subirats, P., Vasseur, P.: A survey of vision-based traffic monitoring of road intersections. IEEE Transactions on Intelligent Transportation Systems **17**(10), 2681–2698 (2016)
9. Dinh, H., Tang, H.: Development of a tracking-based system for automated traffic data collection for roundabouts. Journal of Modern Transportation **25**(1), 12–23 (2017)
10. Engel, J.I., Martín, J., Barco, R.: A low-complexity vision-based system for real-time traffic monitoring. IEEE Transactions on Intelligent Transportation Systems **18**(5), 1279–1288 (2017)
11. Girshick, R.: Fast r-cnn. In: IEEE International Conference on Computer Vision (ICCV) (2015)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 583–596 (2015)
14. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: IEEE International Conference on Computer Vision (ICCV) (2015)
15. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., ehovin, L., Vojir, T., Bhat, G., Lukei, A., Eldesokey, A., Fernandez Dominguez, G., Garcia-Martin, A., Iglesias-Arias, ., Alatan, A., Gonzalez-Garcia, A., Petrosino, A., Memarmoghadam, A., Vedaldi, A., Muhi, A.: The Sixth Visual Object Tracking VOT2018 Challenge Results, pp. 3–53 (01 2019)
16. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
17. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. arXiv preprint arXiv:1711.07264 (2017)
18. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
20. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
21. Reid, D., et al.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control **24**(6), 843–854 (1979)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)
23. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International Journal of Computer Vision (IJCV), 2013 (2013)
24. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision **75**(2), 247–266 (2007)
25. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision (ECCV) (2014)