Proceedings of the 6th RSI
International Conference on Robotics and Mechatronics (IcRoM 2018)
October 23-25, 2018, Tehran, Iran

Robust Object Tracking Based on Recurrent Neural Networks

F. Lotfi, V. Ajallooeian and H. D. Taghirad, Senior Member, IEEE
Advanced Robotics and Automated Systems (ARAS), Industrial Control Center of Excellence (ICCE),
Faculty of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran.
Email: F.Lotfi@email.kntu.ac.ir, vahid.ajalluian@email.kntu.ac.ir, taghirad@kntu.ac.ir

Abstract—Object tracking through image sequences is one of the important components of many vision systems, and it has numerous applications in driver assistance systems such as pedestrian collision avoidance or collision mitigating systems. Blurred images produced by a rolling shutter camera or occlusions may easily disturb the object tracking system. In this article, a method based on convolutional and recurrent neural networks is presented to further enhance the performance and robustness of such trackers. It is proposed to use a convolutional neural network to detect an intended object and feed the tracker with found image. Moreover, by using this structure the tracker is updated every $'n'$ frames. A recurrent neural network is designed to learn the object behavior for estimating and predicting its position in blurred frames or when it is occluded behind an obstacle. Real-time implementation of the proposed approach verifies its applicability for improvement of the trackers performance.

Index Terms—Robust object tracking, position estimation and prediction, recurrent neural network, collision avoidance

I. Introduction

Tracing the path of a specific object in continuous frames, recalled as object tracking (OT) in the literature, has recently gained much attention due to its vast potential applications. Although OT is a challenging topic on machine vision, it has found its path toward traffic monitoring, video surveillance and autonomous vehicles. There are several methods for object detection (OD) and tracking. Earlier methods of OD was based on feature extraction and edge detection. As in ILSVRC 2012 [1], deep neural network gained attention in machine vision and machine learning area, on which AlexNet [2] was introduced. Thereafter, most of the researchers supposed to use Convolutional Neural Network (CNN) as their main framework. Single stage and double stage methods were used in CNN based object detectors. Double stage methods first look for an object proposal and then recognize objects using region proposals [3]–[5]. On the other hand, single stage methods, create labeled bounding boxes without looking for object proposals and use direct detection methods [6]–[9].

In general, detection takes more time than tracking because of two main reasons. OD algorithms look for all objects on their training data set, but OT algorithms usually look for specific objects chosen on the earlier frames of videos until logically selected objects get out of frame. Furthermore, unlike an object detector that searches for the object in all pixels of a frame, the object tracker uses only neighboring pixels to the previous object pixels in the frames. However, trackers can follow either a single object or they can track multiple objects. Available trackers have major problems on which blurring (caused by using rolling shutters), occlusions, change of illumination and rotations are of most frequent issues. This matters will lead to losing the goal. [10]

Two important measures to qualify these algorithms are performance speed and precision. Algorithm speed usually assessed by frame per second (fps) execution. There are some parameters defined for measuring the precision of these methods. Let us define the total number of objects in a frame as o_t, the total number of object detected by algorithm as o_d, the total number of correct object detected as o_{cd} (which means there was an object, but not necessarily correctly recognized), the total number of correct object recognized as o_{cr} (which means there was an object, and necessarily correctly recognized), total wrong object detected as o_{wd}(which means there was no object, but algorithm recognized that box as an object) and the total number of the wrong object recognized as o_{wr}(which means there was an object, but algorithm wrongly recognized that box). Ratios of these parameters provide useful measures for precision evaluation and comparison. The most common measure that is used as a metric for performance evaluation is the mean average precision [11].

Object trackers have different roots. Conventional systems can be grouped into two main classes, correlation, and non-correlation filter trackers. The simplest method, Kalman filter, is based on correlation filtering. Some of the methods used color histograms, which can be executed quite fast. Non-correlation algorithms may also be classified into conventional and learning approaches. Since CNNs has a very well record in detection problems, researchers tend to use this tool for the purpose of tracking, as well [12].

In this paper, a method based on convolutional and recurrent neural networks is proposed to further enhance the performance and robustness of the trackers. It is proposed to use a convolutional neural network to detect and find an intended object and feed the tracker with it. A recurrent neural network is designed to learn the object behavior to estimate its position in blurred frames and

978-1-7281-0127-9/18/$31.00 ©2018 IEEE

even when it is occluded behind an obstacle. Realtime implementation of the mentioned approach verifies its applicability in improvement of the trackers performance.

The rest of this paper is organized as follows: In section II the backgrounds are addressed. Section III focuses on the combination of detector and tracker. Moreover, the new method on object tracking is going to be introduced in this section. In section IV experimental results are given. Finally, concluding remarks and future work are presented in section V.

II. Required Backgrounds

On the search for autonomous robots, researchers proposed many platforms with different sensors such as sonar, LIDAR and optical sensors. These sensors provide big data and information taken from these data is needed for the control of a robot. One of the most used sensors in this area is the image sensor. Data gained from sensors can be processed either in real time (for autonomy) or in long time (for diagnosis). During the last two decades, various methods have been proposed with different hardware used for processing. Albeit, fast detection in frame sequences known as videos are very useful to show algorithm performance, it can not be used for decision-making as one of the important parts of autonomous robot software. Here object tracking is what makes the basis of this goal. Trackers can trace either a single object or multiple objects in videos.

A. Detection

Humans can identify objects by watching them. Until today, it is not known what will happen that a human can distinguish different things to imitate for machines. First techniques of object detection were based on pixel-wise approaches like edge-detection and feature extraction. Having suitable functionality on limited objects, these methods could be updated from one object to another while this procedure takes a long time, to modify them for detection of other objects. The problem with these methods is that they are not robust against rotation and change of size, which was solved using SIFT [13] and SURF [14] algorithms. However, their solution was for in-page transformations and spatial transforms were still unsolved until CNNs gained power.

One of the earlier networks, LeNet [15] was used for recognition of small size picture of digits. CNNs were not strongly investigated until powerful hardware were developed. On 2012 Imagenet challenge, AlexNET influenced machine vision and machine learning techniques which resulted in the broad use of CNNs. Based on the context of the convolutional neural network, researchers developed some state of the art architectures. Girshick et al. [3] proposed RCNN on which first they look for object proposals, and then a recognition performed on the offered bounding box. Given precision was good, however, the speed was not satisfactory. On the proceeding of RCNN, Girshick upgraded image proposal creation speed and established Fast R-CNN [4]. At the same time, Redmon et al. proposed YOLO [6], which on the contrary to RCNN, no box proposal suggestion is used and the whole work was on CNN. The image was sliced into $N \times N$ grids and for each grid, the network produced B bounding boxes and calculate the probability of class i existence in that box. In the meantime, another verification parameter, named confidence is calculated, by which the probability of an object being exist in that bounding box is evaluated.

Despite demonstrating sufficient accuracy and precision, the speed of producing bounding box and labeling them still needs to be upgraded. On the way to reach real-time object detectors, Redmon et al. improved their previous work and proposed YOLO 9000 [7]. Furthermore, Ren et al. in Faster RCNN [5] followed their preceding endeavor, shared CNN outputs inside different parts of their architecture and improved detection speed. Another single stage network like Yolo is also proposed at the same time, which is called SSD [9]. On SSD, Liu et al. used the Multibox [16] approach, but with fixed size priors. Nowadays, with the advances in semiconductors, there are some powerful hardware that support running and training of these networks. In this paper we supposed to use the YOLO [8] for detection.

B. Tracking

Although detectors provide information on the location of different objects, however, even video detectors cannot provide any information about kinematics of the environment and motion model of objects inside a frame. In order to control a robot, obstacle collision avoidance and navigating on the road, robot software should be able to handle the aforementioned problems. The very first step to implement the software is to develop a tracking algorithm. Efforts to build an object tracker goes back to the same time when the detectors were introduced in the literature.

As Kalman filter is an acceptable approach in control and communication theory, it seems apparently worthwhile to implement a two-dimensional filter or two one-dimensional filters on the video scenes to catch the object in next frames. Furthermore, there are other trackers, like dense and sparse optical flow, in which by changing the optical flow and difference of present and past frames, the motion of objects will be depicted. In addition, there are some old probability-based approaches like Camshift and Meanshift methods, that were used [17]. Moreover, the fuzzy framework is also used for this target [18]. Following CNNs success in the detection task, it was suggested that they may be used for single object tracking [12].

The approach will be naturally different whether the goal is to track a single object or multiple objects. Newer designs can select the type of tracking and optimize it. For example, multiple instance learning has been introduced in [19] is based on different instances of an object and considering the idea of tracking by detection.

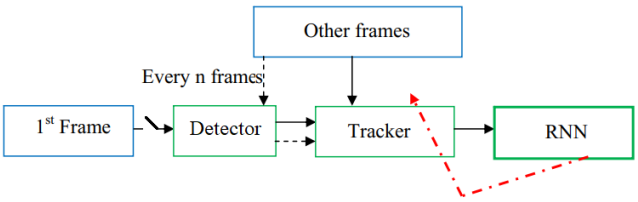


Fig. 1: Block diagram of the proposed tracker

III. Combining detector and tracker

Amongst all the previously proposed approaches, there is a major shortcoming when the object is not detectable because of occlusion or image blurring caused by fast motions, which leads to losing the target. Since designed methods are based on the image itself, and not the trend happening inside the video, this problem seems to be inevitable, regardless of what method is used for tracking purpose. Furthermore, it is necessary to first identify which object shall be tracked in an autonomous tracker.

To solve this issue, YOLO detector is used to find the predetermined object for the first time, where it is assumed that the image is not blurred, to get the initial position of the object. Thus, after manually setting the desired object, the tracker is ready to seek the goal. As soon as the search is finished, the object location will be used for the tracker to trace the goal. Due to the high speed needed for the tracker, it is not possible to use the detector for tracking and use the detector system on each frame. In this framework, normal movements of the object are fully detectable. However, to solve the problem of occlusions, the detector should run every n frames. By these means, the tracker may be easily implemented in realtime by using a detector scheme. However, the problem of blurred images still remains unsolved.

In this paper, it is proposed to use an RNN to predict and estimate the position of the object even when the detector is failed to update the tracker. The inherent characteristic of RNNs is that they have memory and in contrast to the other methods they use the existing dynamic inside the data. In the case of blurred images and occlusions, the designed RNN helps tracker in following the object even behind an obstacle, if the object does not change its path during the occlusion. It is proposed to train the RNN every m frames (for instance, every 50 frames). In essence, RNN is updated with the recent object motion pattern.

The designed RNN structure consists of two layers, with three inputs, two outputs and five neurons in the mid-layer. Inputs are the three consecutive samples of the center of the object's bounding box position. The network is trained for 1000 epochs. Learning rates for the first and second layer are 0.01 and 0.1, as well. The important point is that due to the low amount of data which is obtained from the few frames, for the proper training of the RNN, data are shuffled for each epoch. By this means, the robustness of the neural network is increased. Another point to note is that in case of slow motions, the training of the recurrent neural network is practically difficult but it causes no problem. Since, in low speed cases, the image is not going to be blurred and it is not necessary to use this network.

Figure 1 shows the block diagram of the proposed approach, in which on the first frame we mean the frame which the detector has found the object on the first time. As it is indicated in this diagram, first the tracker is fed with the detector. By this means, the necessary condition of object existence in the first frame is eliminated. Thereafter, using the detector every n frames the tracker is updated. Finally, in the case of occlusions and blurred images RNN helps the tracker to find the object.

IV. Experimental Results

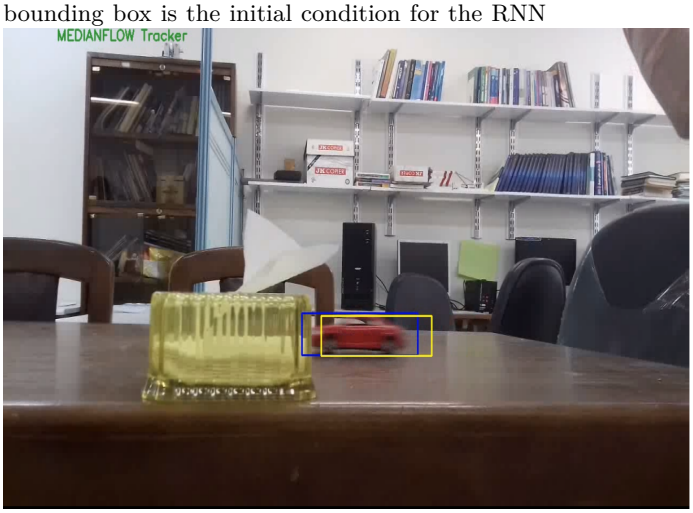
There are several kinds of trackers developed in OpenCV such as BOOSTING, MIL, KCF, TLD, MEDIANFLOW, GOTURN, or MOSSE. For the experiments, we used MEDIANFLOW tracker. The reason to choose this tracker is that this tracker is able to track the object in both forward and backward directions in time and measures the variations between these two. By minimizing this forward-backward error the capability to reliably detect tracking failures and select reliable paths in video sequences is realized. The goal is to find and track a car.

As it is indicated in figure 2a, YOLO detector has successfully found the car and the respective bounding box is shown with the blue color. Yellow bounding box in figure 2b shows the RNN initial condition and the blue one is the MEDIANFLOW tracker output. As discussed in the previous section, we need some initial frames to get the minimum data for the RNN training. It is shown in figure 2b that the tracker is running well because there is no occlusion and the detector is not running on the blurred image. But as the car speed increases, the upcoming frames are going to be blurred. Moreover, an obstacle in front of the car path is considered to evaluate the effectiveness of the proposed algorithm in occlusion.

Figure 2c displays the frame just before the tracker failure. Thereafter, in figure 3a the tracker has failed to track the car as it is hidden behind the obstacle. It is noteworthy to mention that as the MEDIANFLOW fails to track the object, the bounding box position which indicates the tracker result is fixed and it is not moving. Thus, usually the last position of the object according to the MEDIANFLOW tracker output is saved for the blue bounding box. At this time the RNN uses all the data obtained from the time the object was detectable to train itself. Hence, it can estimate the position of the car for the upcoming frames. The reason of using the mentioned inputs for the RNN is to consider a fixed acceleration for the car motion model. In figure 3b we can

(a) First frame that the car has appeared and the blue bounding box is determined using YOLO detector.

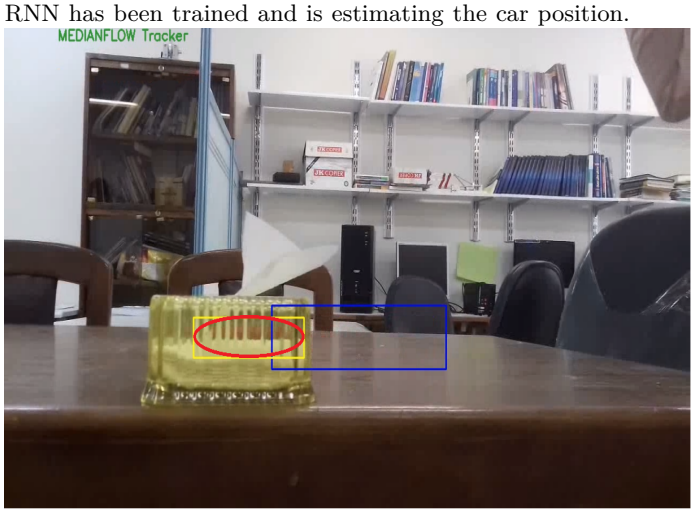
(b) A frame that the tracker is running and the yellow colored bounding box is the initial condition for the RNN



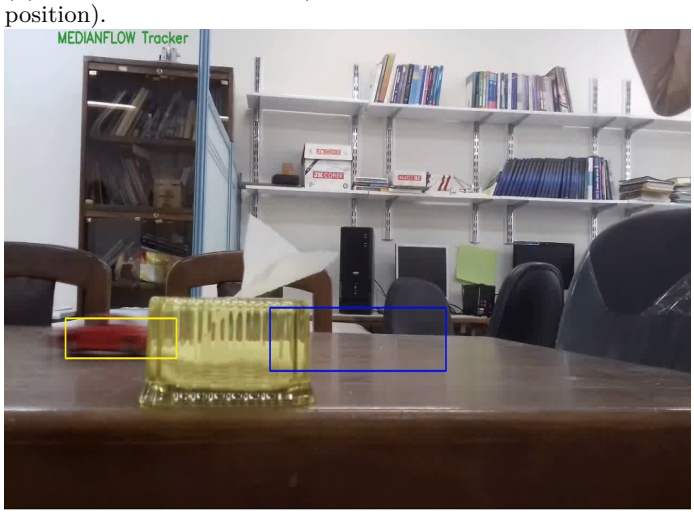
(c) The frame just before tracker failure.

Fig. 2: Sequence of frames before tracker failure

(a) MEDIANFLOW tracker has failed to track the car while the RNN has been trained and is estimating the car position.



(b) RNN result in occlusion(Red region is approximately the car position).



(c) RNN result on a blurred frame after occlusion.

Fig. 3: Sequence of frames after tracker failure

see the appropriate proficiency of the presented approach in tracking the object behind the obstacle. Moreover, figure 3c shows the robust performance of the mentioned method in tracking the object after the occlusion and in blurred frames. As it is seen, the RNN has been well-trained to estimate the position of the car when both detector and tracker have failed to find the car. These results clearly verify the effectiveness of the proposed approach, and the robustness of the tracking performance in presence of blurred images and occlusion.

V. Conclusion

In this paper, a robust object detection and tracker is proposed based on CNNs. Although existing object trackers are much faster than a regular CNN, they are quite fragile in detecting objects behind obstacles, and in blurred images. In this paper, it is proposed to combine CNN with customized trackers, and to use a recurrent neural network structure to estimate and predict the moving object position in fast motions. This method is very promising in collision avoidance autonomous systems. The experimental results show the effectiveness of the proposed method in presence of occlusion and blurred images.

References

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[3] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524, 2013.

[4] Ross Girshick. Fast r-cnn. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[5] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015.

[6] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. CoRR, abs/1506.02640, 2015.

[7] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. CoRR, abs/1612.08242, 2016.

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. CoRR, abs/1804.02767, 2018.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.

[10] S. Siena and B. V. K. V. Kumar. Detecting occlusion from color information to improve visual tracking. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1110–1114, March 2016.

[11] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, January 2015.

[12] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In European Conference Computer Vision (ECCV), 2016.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91–110, 2004.

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Comput. Vis. Image Underst., 110(3):346–359, June 2008.

[15] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, pages 2278–2324, 1998.

[16] Christian Szegedy, Scott E. Reed, Dumitru Erhan, and Dragomir Anguelov. Scalable, high-quality object detection. CoRR, abs/1412.1441, 2014.

[17] John G. Allen, Richard Y. D. Xu, and Jesse S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing, VIP '05, pages 3–7, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[18] Shunli Zhang, Sicong Zhao, Yao Sui, and Li Zhang. Single object tracking with fuzzy least squares support vector machine. IEEE Trans. Image Processing, 24(12):5723–5738, 2015.

[19] Wuzhen Shi, Wenfei Wang, Dianbo Li, Zhizong Wu, and Lin Mei. Visual tracking with online multiple instance learning based on background classification. In James J. (Jong Hyuk) Park, Yi Pan, Cheon-Shik Kim, and Yun Yang, editors, Future Information Technology, pages 409–415, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.