A Robust Long-Term Pedestrian Tracking-by-Detection Algorithm Based on Three-Way Decision

Ziye Wang[1], Duoqian Miao[1,2(✉)], Cairong Zhao[1,2], Sheng Luo[1], and Zhihua Wei[1,2]

[1] Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
{yeziwang, zhaocairong, zhihuawei}@tongji.edu.cn, miaoduoqian@163.com, tjluosheng@gmail.com
[2] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China

Abstract. Pedestrian Detection Technology has become a hot research topic in target detection field recent years. But how to track the pedestrian target accurately in real time is still a challenge problem. Recently deep learning has got the extensive research and application in both target tracking and target detection. However, the tracking effect based on deep learning needs to be improved in the motion blur and occlusion cases. In this paper, we propose a new model that combines the target tracking and target detection and introduce the idea of granular computing to realize high-precision long-term robust pedestrian tracking. In this model, we use a pre-trained tracking model to track the specified object and use the three-way decision theory to judge the color histogram feature and correct the results by the detector. Compared with the separated tracker, our model invokes the target detector to detect the current frame when the tracking result is wrong and the detection result which is the most similar to the target is selected as the tracking result. Experimental results show that our model can significantly improve the tracking accuracy especially in the complex situations, compared with the separated tracker and the detector.

Keywords: Long-term tracking · Tracking by detection · Color histogram · Granular computing · Three-way decision

1 Introduction

Conventional target tracking methods cannot handle and adapt to complex tracking changes, and its robustness and accuracy remain to be improved. It is because that these classical algorithms do not rely on the prior knowledge, but use the method of probability density, manual setting or other methods to detect the moving target directly from the image, and finally locate the interest moving target in each frame of a video.

The first author is a student.

T. Mihálydeák et al. (Eds.): IJCRS 2019, LNAI 11499, pp. 522–533, 2019.
https://doi.org/10.1007/978-3-030-22815-6_40

On the contrary, the methods based on deep learning [1] could learn the valid features from big data automatically, which would cost years for conventional methods. That is the reason why deep learning methods are superior to the conventional methods [2] which use the hand-designed features such as HOG [3] or CN in the representation of the feature.

Deep learning has long been used in various fields of computer vision, such as image classification, target detection, semantic segmentation and so on. Until recently, with the great development of big data and the continuous improvement of computing power, deep learning began to be applied in the fields of target tracking and target detection.

In 2013, Wang and Yeung [4] proposed the use of the stacked denoising autoencoder (SADE) to extract the target features from a large number of data by unsupervised pre-training, and then use the particle filter to track online. This is the first tracking algorithm that applies depth models to single target tracking tasks. In 2014, Wang and Yeung proposed SO-DLT [5], which is a successful application of large-scale CNN in target tracking. Long and Shelhamer proposed the FCNT [6], its characteristic is expressing the difference and connection in the target attributes by exploring the CNN features of different layers and using tiny convolution neutral network to make them sparse. These measures can effectively prevent the drift of the tracker and have better robustness to the deformation of the target itself. Nam and Han proposed pre-training CNN by tracking videos directly to obtain general target expression ability and using an innovative multi-domain training method [7]. It wined over other contestants in the VOT-2015 Challenge, it is also the first time that there is alternate training in tracking. In 2016, Held put forward a deep vision tracking algorithm named Generic Object Tracking Using Regression Networks. GOTURN uses offline learning to learn through a large number of video and picture samples [8], so that the network can learn the appearance models and motion models of objects, it is the first time that the target tracking algorithm using depth learning achieves 100 FPS.

At present, the mainstream target detection algorithm is mainly based on the deep learning model. It can be divided into R-CNN algorithm based on region proposal [15], such as Fast R-CNN [16], Faster R-CNN [17], and end to end algorithm, such as YOLO [18], SSD [19]. The Faster R-CNN algorithm is widely used for the moment, which can be regarded as the combination of Region Proposal Network (RPN) [17] and fast R-CNN and uses the idea of shared convolutional layer to reduce the computational burden of proposal generation.

Before GOTURN was proposed, most of the tracking methods based on deep learning cannot meet the real-time requirement [12–15], it is the first model which makes the real time tracking with deep learning is possible. However, the tracking effect of GOTURN algorithm still needs to be improved in the cases of blurred target and target occlusion, and the loss of target often occur. Therefore, we hope to establish a tracking model to achieve impressive success in robust tracking.

Pedestrian is a non-rigid target, the complexity of its scenes and shape changing, view changing makes pedestrian tracking has been a difficult challenge in the field of computer vision research. The tracker could use the information from previous frames to lock the target efficiently but do not work well in complex situations. On the contrary, a well pre-trained target detector could detect the pedestrians easily under the

condition of motion blur or partial occlusion. Therefore, we consider integrating the advantages of tracking and detection and propose a new model which combines detection and tracking to achieve long term robust tracking for pedestrians using the idea of tracking-by-detection.

The pedestrian tracking model built by us includes 3 layers, the first layer is a target tracker used GOTURN algorithm and the third layer is a target detector used Faster R-CNN. The most important second layer is to judge the tracking result and decide whether to call the detector according to the judge result.

The way to judge the result is to calculate the similarity between the two crops of the tracking target and the tracking result. We cannot directly compare two pictures, considering that the target is moving and the shape of the target or the angle of view may change. But at the same time, the color of the target will be roughly the same in a short time. Therefore, we consider using color histogram algorithm [21] to compare similarity between tracking result and tracking target.

In order to improve the accuracy of classification of tracking results, we abandon the traditional two-branch decision and adopt the three-way decision theory in the second layer. In 2009, Yao proposed three-way decision theory [9–11] based on decision rough set theory. Today, three decision-making theories have been widely used in many fields. So far it has been widely used in many fields, such as emotional classification [23] and image classification [24].

In general, our model could improve the accuracy of tracking results efficiently especially in some challenging cases such as fast motion, motion blur and occlusion.

The main contributions of this paper are as follows:

- Construct a long term robust pedestrian tracking model by combining the tracker and detector. Use the detector to correct the tracking result.
- Design an algorithm based on three-way decision to judge the tracking result by the feature of color histogram and decide the appropriate time to call the detector.
- Determine the appropriate threshold to get accurate tracking model.

The rest of the paper is organized as follows: In Sect. 2, we introduce the related work. In Sect. 3, we describe our model. In Sect. 4, we present the experiments and analyze the results. We make the conclusion in Sect. 5.

2 Related Work

2.1 The Faster R-CNN

Since the concept of deep learning has been introduced in many computer vision tasks, especially target detection, there were a lot of algorithms have been proposed which based on CNN. Both the R-CNN [15] and the Fast R-CNN [16] rely on the CNN to extract the feature form the proposals. But the speed of proposal generation is not good enough. Therefore, to reduce the computational burden of proposal generation, the Faster R-CNN was proposed. It could be considered as a combination of the Region Proposal Network (RPN) [17] which could extract proposals quickly and the Fast R-CNN detector whose purpose is to refine the proposals. The most important idea of

the Faster R-CNN is that the RPN share the convolutional layers with the Fast R-CNN, as Fig. 1. In this way, the image could pass through the CNN only once and could extract proposals efficiently.

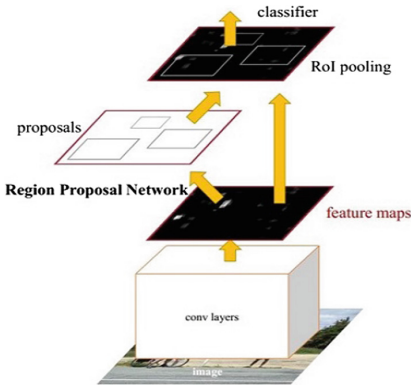

Fig. 1. Network architecture for Faster R-CNN [17].

In the Faster R-CNN model, an input image firstly passes through the Conv layers which was made up of 13 convolutional layers, 13 ReLU layers and 4 pooling layers to extract the feature map of the whole image. Then use the feature map as the input of RPN to get the region proposals. The RoI pooling gathers the proposals and the feature map and to create the proposal feature maps and sends them to the classifier to calculate the class value and process with regression to get the accurate bounding-box.

2.2 Generic Object Tracking Using Regression Network (GOTURN)

There were many algorithms of tracking a single object in a video using deep learning, but the speed of them is too difficult to be assured. So, Held proposed GOTURN which is faster than previous algorithms and can track at 100 FPS [8]. It takes offline pre-training by massive images and videos.

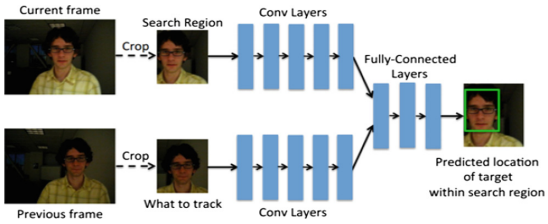

Fig. 2. Network architecture for GOTURN [8].

As seen from Fig. 2, if the target located in the bounding-box centered at c = (c_x, c_y) with a width of w and a height of h in previous frame, it takes two crops of the previous frame and the current frame at c = (c_x, c_y) with a width of $k_1 w$ and a height of $k_1 h$. Then input these crops into the convolutional layers. It supposes that the target object is not moving too quickly and will be located within this region. The outputs of the network are high-level features and are then fed through the fully connected layers. The fully connected layers compare the feature from the current frame and the feature of the previous frame to find where the target has moved.

2.3 Three-Way Decision

In the well-known two-branch decision-making model, only acceptance and rejection are generally considered, but this is often not the case in practical application. Based on the rough set theory proposed by Pawlak [20], Yao's three decision-making theories provide a third alternative to acceptance and rejection: non-commitment [9–11]. The idea of three decision-making is based on three categories: acceptance, rejection and non-commitment. The goal is to divide a domain into three disjoint parts. Positive rules acquired from positive domain are used to accept something, negative rules acquired from negative domain are used to deny something, and rules that fall on boundary domain need further observation, which called delayed decision-making. This way of decision-making describes the thinking mode of human beings in solving practical decision-making problems and has been widely used in decision tree [20] and other fields.

3 The Proposed Algorithm

3.1 Tracking by Detection Model

The tracking algorithms based on deep learning have been deeply investigated in recent years. However, the reality is rather more complicated and the trackers may occur the loss of target in the complex environment. It is because that most of trackers could use the information from previous frame to get the result in the current frame. If the target object moves too fast or it is occluded by other object, the tracker could not get the matching information in the search region and it is likely to lose the target.

In this paper, our research focus on improving the tracking effect in complex situations. For example, our solution of partial occlusion is to introduce a checking scheme based on three-way decision into the model (see as Fig. 3). We use the checking scheme to judge the tracking result to determine if it is occluded, then renew the standard according to the judge result. This method can guarantee the robustness of the standard to the occlusion. In our model, we draw lessons from the idea of tracking by detection. After judging the tracking result, we call the detector when the result is wrong. The detector will get the coordinates of all the pedestrian and select the result which is the most similar to the standard to correct the tracking result. The main notations in this paper are listed in Table 1.

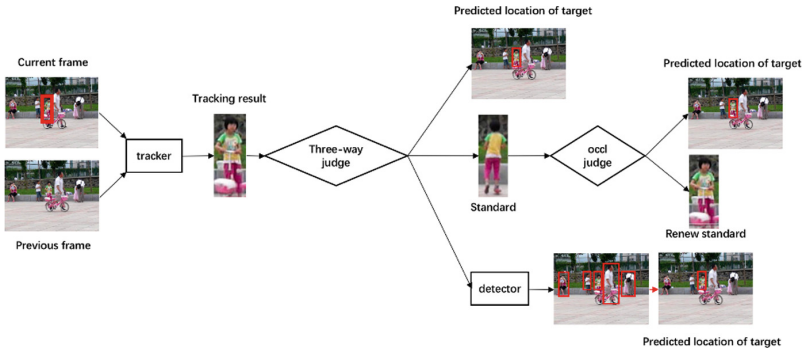

Fig. 3. Framework of our model.

Table 1. List of main notations.

Variable	Explanation
Img_{t-1}	The previous frame
Img_t	The current frame
$c_i = (c_{xi}, c_{yi})$	The center of bounding box of tracking result in frame i
(w_i, h_i)	The width and height of bounding box in frame i
$bbox_i$	The bounding box in the frames i
$crop_i$	The crop at $c_i = (c_{xi}, c_{yi})$ with $(k_1 w_i, k_1 h_i)$
st	The standard of tracking
j	The sequence number of the standard frame
sim_1	The similarity between the $bbox_t$ and $bbox_{t-1}$
$D_t = \{d_{t1}, \cdots, d_{tn}\}$	The detection results of the frame Img_t, including n results
$Th{-}corr$	The threshold to determine if the result is totally correct
$Th{-}wrong$	The threshold to determine if the result is totally wrong
$Th{-}occl$	The threshold to determine if the result is partial occluded
$Th{-}frm$	The threshold to determine if the standard has not been renewed for a long time

3.2 Picture Similarity Discriminant Model

In the long-term tracking tasks, there are many different instances of the tracking result. In the model, we need to make one of three decisions: (a) accept it if it is correct; (b) reject it if it is wrong; (c) delay judgment if it is uncertain. Since the tracker just gives the result which is the most possible, we need to judge the result by calculating the similarity between the tracking result and the tracking target. We set two threshold and judge the tracking result is correct if meeting the condition1: $sim_1 > Th{-}corr$ and is wrong if meeting the condition2: $sim_1 < Th{-}wrong$. The condition1 guarantees that the crop of result in previous frame is similar to the crop of tracking result in the current frame. We define the condition under the hypothesis that meeting the condition means that the tracking result is not wrong (see Algorithm 1). The condition2 guarantees that

the crop of result in previous frame is totally different to the crop of tracking result in the current frame.

However, there is also a situation that the tracking result is not wrong while the target is partial occluded. If this situation has not been considered, it may cause the accumulation of errors and will loss the target a few frames later. Thus, we need to judge the result again which we think is correct by the similarity between the result and the tracking target. We establish a Occl-Judge model. We set a threshold and judge the tracking result is not partial occluded if meeting the condition: $sim_1 > Th_3$. If the tracking result satisfies this condition, we can assume that the tracking result in this frame is almost precisely the same to the tracking result in the previous frame.

Algorithm 1 Picture similarity discriminant

Input: $crop_{t-1}$, the current frame Img_t
Output: $bbox_t$
Input Img_t into the first layer (GOTURN), get $bbox_t$;
Send the $bbox_{t-1}$, $bbox_t$ to the discriminant layer;
Calculate the similarity between $bbox_{t-1}$ and $bbox_t$, get sim_1;
If $sim_1 > Th-corr$ **then**
  get $bbox_t$;
else if $sim_1 < Th-wrong$ **then**
  input Img_t into the third layer (Faster R-CNN), get $bbox_t$ with Formula (1);
else
  put crop c_t into the Occl-Judge model;
end if

3.3 Renew Standard Model

Under the theory of color histogram, we build a tracking by detection model based on the color histogram (see Algorithm 2). The first layer is the tracker used GOTURN, and the second layer includes judging whether the result is correct and whether the result is partial occluded. The third layer is the detector (Faster R-CNN). We put Img_t into the first layer, send the tracking result $bbox_t$ to the discriminant layer. Next, the discriminant layer calculates the similarity and make one of two decisions: (a) the result is correct and put the result to the Occl-Judge Model to renew the standard; (b) the result is wrong and call the detector.

If the result is correct, we set the crop as the bounding-box. If the result is uncertain, we calculate the similarity between $bbox_t$ and st, get the sim_2 (We initialize the st to the $bbox_1$). The model need to compare sim_2 with $Th-occl$ and make one of two decisions: (a) the result is almost the same to the standard; (b) the result may be partial occluded.

If the tracking result is not partial occluded, we renew the standard. If the tracking result is partial occluded, we need to return the tracking result as the final result without renewing the standard. But if we do not renew the standard for

Algorithm 2 Tracking by detection

Input: $crop_{t-1}$, the current frame Img_t, st
Output: $bbox_t$
Input Img_t into the first layer (GOTURN), get $bbox_t$;
Send the $bbox_{t-1}$, $bbox_t$ to the discriminant layer;
Calculate the similarity between $bbox_{t-1}$ and $bbox_t$, get sim_1;
if $sim_1 > Th-corr$ **then**
  get $bbox_t$;
else if $sim_1 < Th-wrong$ **then**
  input Img_t into the third layer (Faster R-CNN), detect and get $D_t = \{d_{t1}, d_{t2}, \cdots, d_{tn}\}$
  get $bbox_t$ with Formula (1);
else
  Calculate the similarity between $bbox_t$ and st, get sim_2;
  if $sim_2 > Th-occl$ **then**
    $st = bbox_t$
  else
    if $t - j > Th-frm$ **then**
      $st = bbox_t$
    end if
  end if
else
end if

a long time, that may be the object has moved too quickly or the angle of the view changed a lot. We need to force the model to update the standard.

If the result is wrong, we believe we have lost the tracking target. Then we input the Img_t to the detector layer and get all the detection results of pedestrian. We calculate the similarity between d_{ti} and st, get the sim_{ti}. We will choose the final result from D_t. Model computing see Formula (1).

$$i = index(\max(d_{t1}, d_{t2}, \cdots, d_{tn})) \tag{1}$$

We will choose the detection result which is the most similar to the standard. Return the ith detection result as the final tracking result.

To sum up, our model could handle some special cases and can avoid losing of the target efficiently.

4 Experiments

4.1 Training

In this paper, we do experiments on a deep learning framework named Caffe. We train the Faster R-CNN detector using the Caltech Pedestrian Dataset and the Pascal VOC2007. The Caltech Pedestrian Dataset consists of approximately 10 h of 640 × 480 30 Hz video taken from a vehicle driving through regular traffic in an urban environment. It has about 250000 frames with a total of 350000 bounding boxes. The Pascal VOC2007 dataset has 20 classes. It contains 9963 images and 26460 bounding boxes. We only use the labels of pedestrian in the dataset to train our model. We choose the VGG16 model and set the number of iterations to be [80000, 40000, 80000, 40000]. The mAP of the detector is 0.763 when use the Caltech Pedestrian Dataset. The mAP is 0.777 when use the Pascal VOC2007. And the further experiment shows that the model trained by the Pascal VOC2007 perform better in the deformed cases. Therefore, we select the Pascal VOC2007 pedestrian part to train the detector for the subsequent applications.

4.2 Test Set

Our test set consists of the 16 videos from the OTB-100 Dataset [10] and 14 videos from the VOT 2015 Tracking Challenge [22]. Both of the OTB-100 and the VOT 2015 are standard tracking benchmarks that allow us to compare our tracker to a wide variety of other trackers. All the videos we selected take pedestrian as tracking targets. The trackers are evaluated using two standard tracking metrics: precision and success rate, which range from 0 to 1.

Each sequence is annotated with a number of attributes and we mainly focus on occlusion and motion blur. The trackers are also compared with each other for separately from these two attributes.

4.3 Results

In order to select completely correct and incorrect tracking results from the three decision-making branches, experiments show that when the similarity of two croppers of person exceeds 0.95, they are basically identical, and when the similarity is less than 0.75, they are different. So in this experiment, we set $Th{-}corr$ 0.95, set $Th{-}wrong$ 0.75 and set $Th{-}occl$ 0.98. We use the tracker of GOTURN, MDNet and our model to get groups of the bounding box coordinates. Then we get the coordinates of the ground truth boxes from the corresponding data.

Calculate the center location error of every frame in the test dataset. Define the center location error as the average Euclidean distance between the center locations of the manually labeled ground truth box and the center locations of bounding box in every frame. This metric can show the performance for the whole sequence. The precision is defined as the percentage of frames whose center location error is less than the location error threshold. It can evaluate the overall tracking performance.

Calculate the bounding box overlap of every frame. The overlap is the value of the intersection of bounding box and the ground truth box divide the value of the union of these two areas. The success rate, defined as the percentage of the number of the frames whose overlap is not less than the threshold could measure the performance of the video.

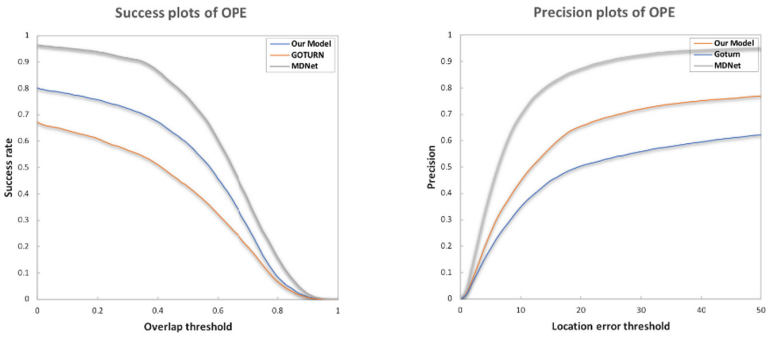

Fig. 4. Success rate plot and precision plot for all 30 sequences. Best viewed in color.

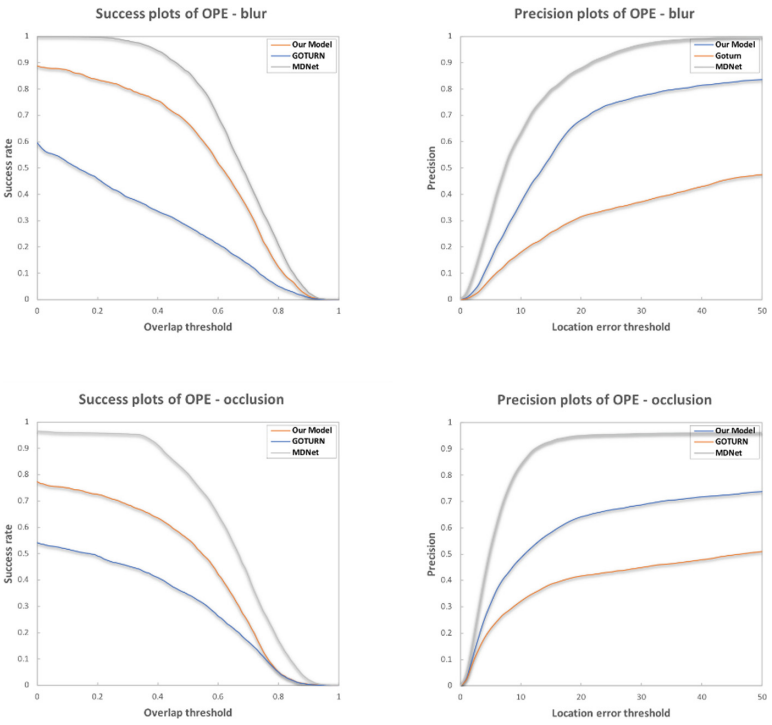

Fig. 5. Success rate plot and precision plot for sequences with attributes: occlusion, motion blur. Best viewed in color.

As can be seen in Fig. 4, both of the success rate and the precision of our model is higher than GOTURN algorithm but lower than MDNet. In the calculation results we can see that the precision of our model is 0.655 when we set the location error threshold = 20. And the success rate of our model is 0.5891 when the overlap threshold = 0.5. This suggest that certain improvement in the overall performance of our model is observed but there is still room to growth.

The sequences in the test dataset are annotated with attributes. It can be seen from the attributes that what challenges the trackers will face in the sequences [9]. According to the purpose of establishing a long-term robust tracker, we report results for two attributes in Fig. 5: occlusions, motion blur.

When we set the location error threshold of occlusion = 20, we can see that the precision of our model is 0.641 while the precision of GOTURN is 0.416. The success rate of our model and GOTURN are 0.556, 0.445 when the overlap threshold of occlusion = 0.5.

When we set the location error threshold of motion blur = 20, the precision of our model and GOTURN are 0.682 and 0.315. The success rate of our model and GOTURN are 0.670 and 0.279 when the overlap threshold of motion blur = 0.5.

Overall, the performance of our tracking-by-detection model is more stable in occlusion or motion blur situations. The experimental results show that our model can effectively improve the accuracy and the stability of long term pedestrian tracking compare with a single tracker based on GOTURN.

5 Conclusion and Future Work

In this study, we integrate target tracking and target detection to build a tracking-by-detection model. The results of experiments show that our model can effectively improve the accuracy of long term pedestrian tracking especially in the cases of occlusion and motion blur. The contributions of this paper are: (i) Judging the tracking result whether it is wrong by using the color histogram. (ii) Introduce the tracking standard and call the detector to make modifications to the tracking result.

In future work, we will do more experiments to adjust the threshold to prove the precision of the tracking and increase the speed of our model.

Acknowledgments. The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions. This work was supported by the National Key R&D Program of China and National Science Foundation of China (Grant No. 61673301). It was also supported by the Major Project of Ministry Public Security (Grant No. 20170004).

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deeplearning. Nature **521**(7553), 436–444 (2015)
2. Danelljan, M., Bhat, G., Khan, F.S.: ECO: efficient convolution operators for tracking, pp. 6931–6939 (2016)

3. Kaaniche, M.B., Bremond, F.: Tracking HoG descriptors for gesture recognition. In: International Conference on Advanced Video and Signal Based Surveillance, pp. 140–145 (2009)
4. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: International Conference on Neural Information Processing Systems, pp. 809–817 (2013)
5. Wang, N., Li, S., Gupta, A., et al.: Transferring rich feature hierarchies for robust visual tracking. Comput. Sci., arXiv:1501.04587 (preprint) (2015)
6. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2014)
7. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking, pp. 4293–4302 (2015)
8. Held, D., Thrun, S., Savarese, S.: Learning to Track at 100 FPS with Deep Regression Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 749–765. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_45
9. Yao, Y.: Three-way decision: an interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS (LNAI), vol. 5589, pp. 642–649. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02962-2_81
10. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Inform. Sci., 341–353 (2010)
11. Yao, Y.Y.: Two semantic issues in a probabilistie rough set model. Fundamenta Informatieae, Manuscript **108**(3–4), 249–265 (2011)
12. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: CVPR, pp. 2411–2418 (2013)
13. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
14. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. CSUR **38**(4), 1–45 (2006)
15. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
16. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
17. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
18. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Computer Vision and Pattern Recognition, pp. 779–788 (2016)
19. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Pawlak Z.: Rough sets. Int. J. Comput. Inform. Sci., 341–356 (1982)
21. Zivkovic, Z., Kröse, B.: An EM-like algorithm for color-histogram-based object tracking. In: Computer Vision and Pattern Recognition, pp. 798–803 (2004)
22. Kristan, M., et al.: The visual object tracking vot2015 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, pp. 1–23 (2015)
23. Zhang, Y., Miao, D., Zhang, Z.: Multi-granularity text sentiment classification model based on three-way decisions. Comput. Sci., 188–193 (2017)
24. Li, X., Zhang, Q.: Image classification algorithm based on tolerance granular model and three-way decisions. Comput. Technol. Autom., 93–96 (2014)