Do tracking by clustering anchors output from region proposal network

Yanke Wang¹, Qidan Zhu^{1,a}, Wenchang Nie¹, and Xiao Hong¹

¹Harbin Engineering University, College of Automation, Institute of Intelligent Control, 150001 Harbin, China

Abstract. Most existing clustering algorithms suffer from the computation of similarity function and the representation of each object. In this paper, we propose a clustering tracker based on region proposal network (RPN-C) to do tracking by clustering anchors output by region proposal network into potential centers. We first cut off the second part of Faster RCNN and then cast clustering algorithms in feature space of anchors, including K-Means, mean shift and density peak clustering strategy in terms of anchors' centroid and scale information. Without fully connected layers, the RPN-C tracker can lower the computational cost up to 60% and still, it can effectively maintain an accurate prediction for the localization in next frame. To evaluate the robustness of this tracker, we establish a dataset containing over 2000 training images and 7 testing sequences of 8 kinds of fruits. The experimental results on our own datasets demonstrate that the proposed tracker performs excellently both in location of object and the decision of scale and has a strong advantage of stability in the context of occlusion and complicated background.

1 Introduction

Visual object tracking, an important member of computer vision, has been attracting researchers for its various applications such as on video surveillance, human-computer interaction, and autonomous driving^[1-3]. The way to do tracking has changed greatly with booming of numerous advanced technologies in recent decades, particularly with the help of deep learning, but it remains some very challenging and unresolved problems. By using convolutional neural network (CNN) as descriptor, some proposed object detection networks bring benefits to tracking task not only in the accuracy but also in the robustness particularly in the context of complex background. Among them, the most commonly used networks come from RCNN^[4], Fast RCNN^[5], Faster RCNN^[6], and Mask-RCNN^[7]. These CNN-based networks can tell the localization and classification simultaneously during tracking, but tracking with them only may result in some objects missed in a few frames and bring a troublesome shortage, i.e. the inefficiency.

In our proposed tracker, we considered to tackle these two issues by (i) using only region proposal networks (RPN), the part of Faster RCNN, to generate a great many of anchors with rich centroid and scale information; (ii) leveraging adaptive clustering algorithms on the output anchors to find the potential centers (including centroids and scales). Specifically, we combined RPN and clustering together called RPN-C to replace the left part of Faster RCNN, as shown in Fig 1. To begin with, a model is built on VGG16 for each object and then RPN plays a role of outputting candidate boxes, the anchors. Afterwards, our RPN-C tracker compared 3 clustering

algorithms cast on the anchors, including K-means^[8], mean shift^[9] and density peak clustering^[10]. Finally, we established a dataset of 8 kinds of fruits containing over 2000 training images and 7 testing sequences with consideration of illumination, scale changing, camera shaking, multi-targets and etc^[11]. We trained models both on the full Faster RCNN and on the part of it and then carried out extensive experiments on testing sequences. The result demonstrated that our RPN-C trackers could maintain the accuracy of tracking and reduce the running time at the same time.

Figure 1. Overall structure. The feature extracted from CNN is fed into RPN to get the information of anchors followed by two cases. In initialization part, we use Fast RCNN to initialize the boxes and in tracking procedure, we cast clustering algorithms to find the centers of boxes.

2 Methodology

* Corresponding author:^a Qidan Zhu: robertwang1994@gmail.com,

2.1 Initialization network

In initializing part, Faster RCNN inputs a fixed-height image for feature extraction by VGG16 and then the output is fed into RPN network to obtain the boxes offsets and scores of being object or not at every position in the feature space. According to the information of RPN, we could obtain rich proposals of locations for targets and scale them into feature size to reshape the feature of VGG16 into batches with the same size. All the batches will be fed into FC layers to give the final result of centroids and scales. Afterwards, we consider all the result of first several frames to give the more correct initialization by voting (Fig 1).

After Initialization, we need to maintain the tracking procedure using part of Faster RCNN (VGG16 and RPN) combined with clustering. Compared with clustering in the image space, clustering in subspace of feature has great advantages of efficiency and accuracy. For one thing, subspace of one image is much smaller than the original image in size, which only needs a small number of iterations to achieve the whole process. For another thing, the replacement of pixels with anchors and their information in the subspace extracted by RPN is much more accurate than the color or texture space in the original algorithm.

2.2 Modified clustering based on region proposal network

2.2.1 Mean shift and K-means

Clustering algorithm is a kind of density estimation based on some kernel function convolving data into centers^[8-9]. The key idea of mean shift is to estimate the density by computing means with kernel function. Assuming that finding centers both for the centroids and the scale of boxes has the same operation, we let $\{C^i\}_{i=1}^{t-1}$ denote all

centers of one object over time, where C^i is the decided center at frame i and contains two main components (centroid, scale), i.e. $C^i = \{X^i, Y^i, W^i, H^i\}$. For the candidates of predictions at frame t , similarly, $P_{1:k}^t = \{P_1^t, P_2^t, \dots, P_k^t\}$ denote k candidate regions with 2 components as well, where $P_j^t = \{X_j^t, Y_j^t, W_j^t, H_j^t\}$.

Given predicted center P_j^t , the next center can be computed by equation below

$$P_{j+1}^t = \frac{1}{n} \sum_{i=1}^n K(P_j^t - P_i^t) \quad (1)$$

where $K(*)$ is the Gaussian kernel function.

More exactly, our data to be clustered is the rich anchors output from RPN with 4 dimensions and the update under estimation is mean vector and shift vector. Thus, the centers can be found in two styles, one of which is to gather the anchors as one 4-dimension vector considering centroids and scales simultaneously and the

other of which is to regard anchors as two 2-dimension vectors and to cluster them for centroids and scales independently. In practice, the number of boxes to be clustered is known and initialization of clustering can be replaced with last positions. In this case, this mean shift algorithm could be regarded as K-means algorithm with fixed initial number of centers. In our work, both two kinds of algorithms are considered to do comparison.

2.2.2 Density peak clustering

Besides mean shift, a more popular and attractive clustering algorithm^[10] allows us to do the tracking in a more novel way. This algorithm was defined with the assumptions that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. For the anchor information, every center with higher density of votes can be considered as the object location. To this end, we compute two quantities for each anchor P_i^t : its local density ρ_i^t and its distance

δ_i^t from ones of higher density, both of which depend only on the distances d_{ij}^t between anchors. The local density ρ_i^t of data point—each anchor P_i^t is defined as $\rho_i^t = \sum_j \chi(d_{ij}^t - d_c)$, where d_c is the cut-off distance

and $\chi(x)$ is the kernel function defined as below

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Besides, δ_i^t is measured by

$$\delta_i^t = \min_{j: \rho_j^t > \rho_i^t} (d_{ij}^t) \quad (3)$$

The strategy is cast only to find the global and local maxima on the basis of distance between anchors but simpler than the original algorithm, the assigned members need not to be selected, bringing a much more efficient procedure. Again, the centers of centroids and scales can be clustered together or independently.

3 Results and discussion

Our proposed algorithm was evaluated on 7 sequences in our own dataset^[11] in terms of overlap ratio (OR), center location error (CLE) and operating speed (OS). These 7 sequences are divided into 4 categories according to the interference from their interior and background including grape with static background, mango/strawberry with multi-object information, apple/pear with camera shaking and orange/banana/kiwifruit with change of scale in the context of occlusion. In the tracking part of our RPN-C trackers, 3 kinds of clustering algorithms were considered to do comparison including so-called K-means (KM), mean-shift (MS) and density peak clustering (DP). In addition, we run the tracking by clustering anchors in 2

fashions for centroids and scale both simultaneously and independently (S and I). In detail, we implemented our tracker in python on an Intel E5-2620 2.10 GHz CPU with 8 GB RAM and a GeForce GTX 1080Ti GPU.

3.1. Evaluation of three factors

As for the first two performance factors, CLE is calculated as the error in pixels between the center of the ground-truth bounding box (GT) and predicted bounding box given by the tracker (PRE).

$$CLE = \sqrt{(GT_x - PRE_x)^2 + (GT_y - PRE_y)^2} \quad (4)$$

while OR is calculated as

$$OR = \frac{area(GT \cap PRE)}{area(GT \cup PRE)} \quad (5)$$

Table 1. CLE of six trackers compared with Faster RCNN.

Sequence	KMS	KMI	MSS	MSI	DPS	DPI	Faster RCNN
grape	84.24	64.03	64.20	44.98	48.10	40.28	237.78
mango/ strawberry	39.52 59.64	70.63 61.77	60.90 33.48	34.40 33.41	91.20 34.56	70.32 90.26	196.84 255.13
apple	140.78	165.48	48.88	70.68	161	98.6	49.16
pear	102.05	222.47	133.18	140.95	176.43	142.6 9	107.56
banana	77.36	75.55	50.09	49.03	146.47	76.49	256.23
orange	71.17	34.31	50.02	53.13	29.55	74.49	192.11
kiwifruit	50.46	49.95	41.88	42.98	25.44	121.7 8	225.14
average	78.15	93.02	60.32	58.69	89.09	89.36	189.99

Table 2. OR of six trackers compared with Faster RCNN.

Sequence	KMS	KMI	MSS	MSI	DPS	DPI	Faster RCNN
grape	0.49	0.50	0.51	0.53	0.55	0.59	0.30
mango/ strawberry	0.64 0.29	0.41 0.23	0.55 0.46	0.47 0.44	0.42 0.52	0.40 0.26	0.37 0.26
apple	0.27	0.20	0.42	0.38	0.22	0.27	0.62
pear	0.24	0.07	0.21	0.22	0.20	0.21	0.52
banana	0.41	0.41	0.57	0.34	0.29	0.42	0.28
orange	0.39	0.45	0.25	0.33	0.44	0.23	0.39
kiwifruit	0.42	0.42	0.40	0.43	0.53	0.16	0.33
average	0.39	0.34	0.42	0.39	0.40	0.32	0.38

Table 3. OS(ms) of six trackers compared with Faster RCNN.

Sequence	KMS	KMI	MSS	MSI	DPS	DPI	Faster RCNN
grape	59	59	133	177	190	255	184

mango/ strawberry	59	62	120	164	179	239	184
apple	59	60	79	107	83	94	187
pear	57	60	71	82	64	67	188
banana	58	58	102	139	130	162	184
orange	58	59	92	103	88	102	189
kiwifruit	59	60	81	87	69	75	183
average	58	60	97	123	115	142	186

Tables 1 and 2 show that the average CLE and average OS for our tracker across the range of 7 testing image sequences. Generally, our RPN-C trackers have better performance in the location of objects and the running speed and maintain the accuracy of the boxes and scales. In terms of all these three evaluation factors, RPN+MS stands out in the simultaneous way but other trackers work excellently in some special occasions. Taking DP clustering as an example, it performs well in the context of scale change and occlusion such as kiwifruit and orange. If we compare the ways to do one single clustering, we can see no significant difference between clustering centroids and scales together and independently, but it has much to do with the operating speed. Thus, we could use the RPN-C trackers in the simultaneous way but add some bias to centroids and scales when computing the distance between anchors. To compare them with Faster RCNN to do tracking directly, they have a strong advantage of running speed, apparently the result of the cutting of the FC part. If running the tracker using only CPU, we can get a more impressive comparison. As the performance in tracking accuracy, owing to the missing boxes of Faster RCNN, it gets a bad score since we set a high value if no boxes are detected. Nonetheless, if Faster RCNN gives one box with a very high confidence, the accuracy of both centroids and scale will be very high. However, it loses boxes frequently and detects some fake boxes with lower robustness, resulting in its bad general performance.

3.2 Visualization of tracking

For a more vivid visualization, we select the best sequence from every occasion—apple from camera-trembling situation, grape from color change, kiwifruit from scale change and existing occlusion and mango/strawberry from multi-object occasion—to show their tracking results in images. To be exact, the boxes detected by these trackers are shown in different colors. The cyan boxes are detected by Faster RCNN itself and in these four occasions, we chose RPN+MS+S, RPN+DP+I, RPN+DP+S and RPN+KM+S respectively marked in different colors. As Figure 2 shows, many boxes are missed by Faster RCNN especially when the targets have bad appearance. In fact, the testing sequences were collected one year after the training set using a different camera, so they have a big difference between each other particularly in the illumination. That is to say, our RPN-C trackers have stronger robustness to do tracking.

Figure 2. All these pictures show the boxes detected by RPN+MS+S(light blue), RPN+DP+I(purple), RPN+DP+S(dark blue) and RPN+KM+S(red) compared with Faster RCNN(cyan).

5. R. Girshick, "Fast r-cnn," in International Conference on Computer Vision (ICCV)
6. S. Ren., K. He, "Faster r-cnn: Towards real-time object detection with region proposal networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 1137–1149
7. K. He, G. Gkioxari, "Mask r-cnn," in IEEE International Conference on Computer Vision (ICCV)
8. C. Hua, H. Wu, Q. Chen, Toshikazu Wada. K-means Tracker: A General Algorithm for Tracking People. JOURNAL OF MULTIMEDIA ,1(4),46-53,2006
9. R.Collins, "Mean-shift tracking."
10. A. Rodriguez, A. Laio, "Clustering by fast search and find of density peaks," Science 344, 1492 (2014)
11. https://drive.google.com/file/d/1FsCJ_p-saKaA9e2urjequUOEJV26EpkS/view?usp=sharing

4 Conclusion

This paper presents state-of-the-art RPN-C trackers with combination between part of Faster RCNN and clustering algorithms, which produce very promising results on our testing sequences in terms of CLE, OR and OS. This algorithm divides the whole tracking procedure into two parts including initialization of bounding boxes using the full part and the main tracking task using part of Faster RCNN combined with clustering. The primary contribution of this paper is to use modified clustering algorithms to do selection of anchors output from RPN. As for the evaluation, we compared 3 related algorithms both in simultaneous and independent ways for centroids and scales to show their performance. The evaluation results show they outperform Faster RCNN owing to their stronger robustness. In addition, as the DP clustering performs not well as we expected, we need to modify the way to find the global and local centers and cast a new strategy to exploit the local density in the future.

References

1. A. Yilmaz, O. Javed, "Object tracking: a survey." ACM Comput 38(4), 1–45(2006)
2. A. W. M. Smeulders, D. M. Chu, R. Cucchiara, et al, Visual Tracking: An Experimental Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36, 1442-1468 (2013)
3. Y. Wu, J. Lim, "Online object tracking: a benchmark," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2411–2418
4. R. Girshick, J. Donahue, "Rich feature hierarchies for accurate object detection and semantic segmentation," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)