A tracking method combines with detection algorithm based on deep separable convolution network

Yanni Shi¹, Xun Liang¹
¹Beijing University of Technology, Beijing, 100124, China
371351121@qq.com

Abstract. With the emergence of a large number of artificial intelligence technologies, deep learning has become the key technology in the field of computer vision. Object tracking is one of the most important technologies in the field of computer vision. Thus we studied about tracking algorithms and proposed a method mainly hopes to solve the occlusion problem in complex tracking scene. Using object detection algorithms based on deep learning to increase the speed of associations and improve tracking effect. It can return the position of the tracking object unsupervised. Then extract features to store in features library, so that the prediction of trajectory whose features can highly be matched is more accurate and the associations are more reliable. Experiment shows our tracking algorithm combines with detection algorithm based on depthwise separable convolution networks not only has a smaller and faster model but also achieved a robustness and real-time tracking in scene where objects are under occlusions.

1. Introduction
An association method between the detector and the tracker is designed in our paper. The detector is used to extract the object features and initialize the position of the object. The tracker is used to predict the possible position of the tracking target at the next moment. In order to reduce the interference of occlusion and background factors, we add an apparent feature confidence, using object detection algorithm to return the extracted features and the initial position of the tracking target. Add the cosine distance and Mahalanobis distance to comprehensively consider the most likely position of the tracking object at the next moment, and finally update the tracking track. At last update the parameters of the tracking system for next tracking.

Experiments show that the proposed tracking algorithm based on deep separable convolutional neural network can track the objects in complex background better and has certain robustness in motion blur, occlusion and scale changes. Quantify the model to reduce its volume so that to achieve real-time operation on embedded devices such as mobile phones.

2. Object detection algorithm based on deep separable convolution network
This chapter uses a deep separable convolution structure as the basic network for the object detection network. This structure makes the network structure more 'deep' rather than more 'wide', greatly optimizing the volume of the model, and by designing the input of the model, it can extract a larger size of the feature map. Experiments show that large-scale feature maps are more suitable for small object detection, while small-size feature maps are more suitable for large object detection. The improved detection algorithm can detect small objects far away from the camera and large objects closer to the camera, and also improve the recall of object detection which is under the background noise interference.
2.1 Deep separable convolutional structure
The deep separable convolutional neural network decomposes the conventional convolution kernel into a depthwise convolution and a 1×1 pointwise convolution. Such decomposition operations greatly reduce an amount of computation of conventional convolutional networks. Assume that the input feature map has M channels and the size is $D_k \times D_k$. The size of the convolution kernel is $D_k \times D_k$. when the output channel is N, the calculation of the depth separable convolution network is only $1 + \frac{1}{D_k^2}$ of the standard convolution, as is shown in Equation 2-1.

$$\frac{D_k \times D_k \times M \times D_F \times D_F + N \times M \times D_F \times D_F}{D_k \times D_k \times N \times M \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2}$$

In the experiment we used a convolution kernel of which is $3 \times 3$, so the calculation of the depthwise separable convolution is about $\frac{1}{9}$ of the traditional convolutional neural network.

2.2 Detection method based on deep separable convolutional structure
We design the input of the basic network to make the output feature map have a larger width and height. Our basis is that the larger scale of the feature map, the more default boxes generated based on each feature map. At the same time, each cell in the feature map has a smaller receptive field with the image, and the smaller object, such as the object far away from the video camera, can also have a good detection effect. We design the size of the input image of the network to $512 \times 512 \times 3$. It is clear that 3 is the channel of the image, which is the RGB channels. First, using the depthwise separable convolution kernel to extract the input image’s feature maps. We deepen the number of layers in the network in order to prevent the gradient from disappearing. So we need to add a batch normalization layer to each convolution structure to activate value of the previous layer in each batch. It effectively solves the problem of gradient disappearance in deep networks. We only add the activation layer after the previous depthwise separable convolution block, remove the layer after the small dimension output layer, and use the Inverted residual block proposed in MobileNet_V2 to remove the nonlinear activation layer and retain more information. So we can achieve such a better features extraction.

2.3 Train and test
We totally added twelve deep separable convolutional blocks to our pre-network which is used before the detection network. The setting of this value is mainly based on the GPU’s memory, size of input images’, batch size and some other parameters like training time, model size, etc. The changes of losses are shown in Figure 2-1. It can be seen that the loss of the classification and location are calculated simultaneously in the end-to-end detection network.

![Figure 2-1 changes of losses](image)

The experimental result is shown in Figure 2-2:
3. Object tracking technology based on object detection algorithm

3.1 Tracking algorithm design
First, we use the object detection network based on the deep separable convolution structure built in Chapter 3 to detect the initial frame of the video. After initialization, the detection algorithm will return the coordinates of several bounding boxes, at the same time give the confidence. The bounding box with highest confidence of each detected object is selected by using the non-maximum suppression algorithm. Initializing an independent ID number for each detected object and establishing a feature library for it. The feature library is used to store the last three detected boxes which is belong to current object, and maintain the feature library during the period in which the object is possible to be tracked.

During the operation of the tracking algorithm, the detection algorithm runs every 0.6 seconds, and three frames are continuously detected during each operation. A tracking is activated only when these three frames are all detected such object, and then assigned it a tracker. A dictionary is created to manage the ID number of each tracking object and index to the features library of each ID. The dictionary serves as a basis for judging whether to initial a newly-created object ID and an index for updating the existing features library. When the next period the detection algorithm starts, we first use the detected features to match with the existed features in library. Highly matched means such object already exists; if not, it means that object is a new one. So assign an ID number and the value is the next digit of the largest number already in the ID dictionary.

Since it takes a long time to match with the existing object features in the features library, our detection algorithm is set to perform three consecutive frames every 0.6 seconds to determine whether a new object is present. In order to reduce the calculation time and achieve the purpose of real-time tracking.

3.2 Trajectory prediction
After the object initialization is complete, the next step is to track the object which the tracking status are activated. First, the tracker is used to predict the position of the next bounding box of the object, and the initial value of the coordinate is the position of detected bounding box. Assuming that the current time is $t$, then based on the state of the moment on the motion system, the current state is obtained according to formula (3-1):

$$X_{t|t-1} = AX_{t-1|t-1} + BU_t + W_t$$  \hspace{1cm} (3 - 1)
Next we update the covariance of $X_{t|t-1}$ and let the covariance be represented by $P$, as is shown in equation (3-2):

$$P_{t|t-1} = A P_{t-1|t-1} A' + Q$$  \hfill (3 - 2)

The covariance of the object motion system is represented by $Q$. Equations (3-1) and (3-2) complete the prediction of the state of the motion system together. With the predicted value of the state of motion system, we need to read the measured value of the object position according to the object detection system. The current optimal estimate $X_{t|t}$ for time $t$ is as shown in formula (3-3):

$$X_{t|t} = X_{t|t-1} + K_t(Z_t - H X_{t|t-1})$$  \hfill (3 - 3)

After getting the best estimate of the object motion system, we also let the tracker continue to iterate until the end of the tracking state. Therefore, we will continue to update the covariance of $X_{t|t}$ in the current $t$-time state, as shown in Equation 3-4:

$$P_{t|t} = (I - K_t H) P_{t|t-1}$$  \hfill (3 - 4)

3.3 Tracker summary

In this experiment, the improved Kalman filter algorithm is used as a tracker. The motion feature is used to constrain the object position state as well as the apparent feature matching degree. Only position status which satisfy both constraints can be successfully associated with the motion trajectory. The object detector runs every 0.6 seconds during the tracking, and each frame is continuously detected for three frames. Using this three frames to match with the objects in the feature library. If the matching is successful, the features in the object feature library will be update. If it fails, wait until it matches with a new tracking object feature which means the tracking state of the object is activated. Initialize a tracker for it and prepare to enter tracking state.

4. Experiment results and analysis

We use some video sequence in VTB data set to test our algorithm, the test results are shown in Figure 4-1.
As is shown above, in Figure 4-1(a), the human’s ID whose position is returned by the detection algorithm is initialized. There comes with three human targets. The last three frames of each human which detector extracted have been stored, with the same time a dictionary has been established for their ID management. The human target of ID_3 in Figure 4-1(b) went out of sight. So when it comes to the time threshold, this tracking target will be deleted. Due to occlusion, ID_2’s state changed from tracking to be observed. In Figure 4-1(c), ID_2 walked out of the occlusion so that the tracking state is restored by feature matching. At this moment, ID_1 also enters the state to be observed due to occlusion. In Figure 4-1(d), ID_1 went out of occlusion, and the tracking ID is restored according to the association algorithm. This section verifies that our object tracking algorithm based on deep separable convolution structure can track the object with occlusion well for a long time through experiments.

5. Conclusion
In this paper, we propose a tracking algorithm combines with the object detection algorithm based on deep separable convolutional network. It shows that depthwise separable convolution structure can greatly reduce the computational complexity and the model size, so that our model is portable and method is real-time. Using apparent feature matching constraint and trajectory prediction constraint to determine whether to update the object’s location. Experiments show that when tracking object are under occlusion background our algorithm can still quickly and successfully restore its original ID and then realize a robust and real time tracking.

References
[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society.
[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks.
[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., & Fu, C. Y., et al. (2015). SSD: single shot multibox detector.
[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: unified, real-time object detection.
[5] Arasaratnam, I., & Haykin, S. (2009). Cubature kalman filters. IEEE Transactions on Automatic Control, 54(6), 1254-1269.
[6] Kuronen, T. (2015). High-Speed Hand Tracking for Studying Human-Computer Interaction. Image Analysis.
[7] Ma, X., Liu, Q., He, Z., Zhang, X., & Chen, W.S. (2016). Visual tracking via exemplar regression model. Knowledge-Based Systems, 106(C), 26-37.
[8] Olfati-Saber,R. (2010). Kalman-Consensus Filter: Optimality, stability, and performance. Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference. IEEE.
[9] Hsieh, J. W., Peng, C. S., & Fan, K. C. (2008). Grid-based Template Matching for People Counting. IEEE Workshop on Multimedia Signal Processing. IEEE.
[10] Sinha, M., Kaushik, S., Kaur, P., Sharma, S., & Singh, T. P. (2013). Antimicrobial lactoferrin peptides: the hidden players in the protective function of a multifunctional protein. International Journal of Peptides, 2013, 1-12.