A Method of Dynamic Visual Scene Analysis Based on Convolutional Neural Network

Vadim V. Borisov and Oleg I. Garanin(✉)

NRU "MPEI", Moscow, Russia
{vbor67,hedgehog91}@mail.ru

Abstract. In this paper, we analyze the existing methods of Multiple Object Tracking (MOT), point out their advantages and disadvantages. It is noted that the MOT task must be solved together with the detection of these objects, thus developing a method of the analysis of the dynamic visual scene. We propose a method of dynamic visual scene analysis based on the appearance object model. This method allows one to detect images and to get the "deep features" of detection in one Convolutional Neural Network forward pass, as well as to improve the accuracy of tracking objects construction compared to other online methods and perform processing in real time, at the speed of 24 FPS, which is shown experimentally. In addition, the method works both in the conditions of uncertainty and in the conditions of noise detection data.

Keywords: Tracking · Tracking-by-detection · Single Shot Multibox Detector Convolutional Neural Network

1 Introduction

The task of dynamic visual scene analysis is to build the tracks of objects on the input sequence of frames. Thus, the tasks of detection and tracking objects on multiple frames (Multiple Object Tracking) are solved.

To develop a method of dynamic visual scene analysis, it is necessary to solve the problem of object detection, and then perform tracking-by-detection task. The task of tracking objects is most often solved separately from the detection and the joint use of the detector and the tracking method requires first to detect the object, and then to get the features separately for each object, which increases the complexity of processing. For example, in [9, 11], each detection is fed separately to the input of the Convolutional Neural Network (CNN) to get its "deep" features. This approach can improve accuracy, but requires significant computing resources.

In addition, there are quite accurate methods of tracking objects, such as MHT, ELP [5, 8], but they do not allow processing in real time, because they solve the problem of global optimization and require obtaining the entire sequence of frames (off-line methods). Other methods perform real-time processing, but are not enough accurate, for example SORT [1] (online methods).

At the same time, to study the developed methods of tracking, the most often existing datasets offer [6] to work with a large number of false detections and build algorithms that allow to get data from noise. However, while using CNN, it is

S. O. Kuznetsov et al. (Eds.): RCAI 2018, CCIS 934, pp. 60–69, 2018.
https://doi.org/10.1007/978-3-030-00617-4_6

sometimes necessary to deal not with a large number of false positive detections, but with false negative ones.

Thus, the most promising methods for solving problems of recognition and tracking objects today are the methods based on “deep” learning and CNN, because they allow to obtain high accuracy of recognition and tracking compared with other methods.

Therefore, while solving the complex problem of detection and tracking objects, it is advisable to propose a method that allows performing the detection and construction of an appearance model of the object in one pass CNN, under conditions of uncertainty of their detection. The proposed method will improve the tracking accuracy and perform processing in real time.

2 Appearance Object Model and Algorithm for Obtaining “Deep Features” of the Objects Detection

2.1 Model Description

In this paper, the detection of an object in the image is understood as the area of the image, on which the object is selected with the help of a bounding box. While building a method of Multiple Object Tracking Appearance, models that use a certain set of features to describe the object are often used.

Figure 1 shows the structure of the appearance object model using a multiscale detection model. Unlike the well-known solution of [7] Single Shot MultiBox Detector (SSD), this structure is characterized by the addition of a layer of ROI-Pooling to form the “deep features” of detection. The method of tuning a multiscale model for detecting visual objects in CNN is described in [2, 3].

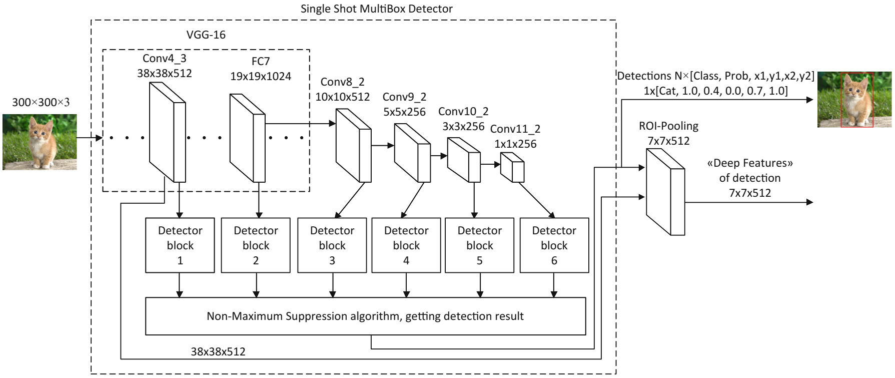

Fig. 1. Structure of the appearance object model

We give a detailed description of the structure of SSD model. This model is based on CNN VGG-16 [10], which contains 5 blocks of convolution layers with a max-pool layer after each block, 3 fully-connected layers and a classifier layer. The first two blocks contain two layers, the other three have three 3 × 3 convolution layers.

To use the VGG-16 as a detector, it is necessary to exclude the last fully connected layer and the classifier one, as well as to include several detection blocks and the layers used to change the scale of the analyzed image. So, one detector block is added to both the feature map of the fourth block (conv4_3) and the fully connected layer (FC7) of model VGG-16. In addition, the network is supplemented by four additional blocks, used to change the scale (conv8_2, conv9_2, conv10_2, conv11_2) with detectors blocks, as shown in Fig. 1. At the network output, the detection results of each block (6 detectors in all) are analyzed with the help of non-Maximum Suppression algorithm and the result of detection is formed.

The structure of the detector block of the Conv9_2 layer is shown in Fig. 2. The rest of the detector blocks are constructed in a similar way.

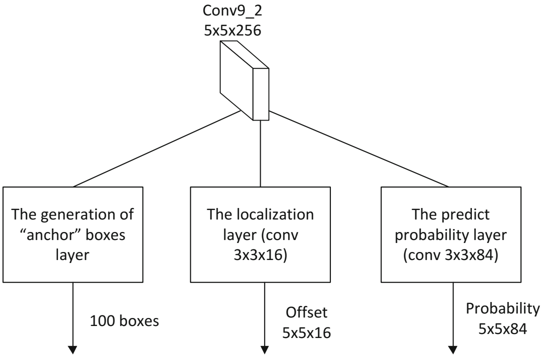
Conv9_2
5x5x256
The generation of "anchor" boxes layer
The localization layer (conv 3x3x16)
The predict probability layer (conv 3x3x84)
100 boxes
Offset 5x5x16
Probability 5x5x84

Fig. 2. Structure of the detector block

The detector block consists of: the generation of "anchor" boxes layer, the localization layer (which predicts the offset of X/Y coordinates from the center relative to the generated box, and the offset along the length and width) and the layer at the output of which the probability that the original box contains an object of the given class. The probability is predicted for each of the classes. For example, if there are 21 classes (20 classes and background) and 5 × 5 × 4 = 100 "anchor" boxes (4 "anchor" boxes for each position of the feature map), the dimension of the output map of the localization layer is 5 × 5 × 16 (4 boxes × 4 offset coordinates) and that of the probability layer is 5 × 5 × 84 (21 class × 4 boxes).

The generation of "anchor" boxes layer constructs "anchor" boxes, the number of which depends on the dimension of the block feature map, on which the detector is built. Such rectangles cover the entire image with a grid, as shown in Fig. 3.

It is important to consider the inputs and outputs of this model. The input gets an input RGB image (frame), which is converted to 300 × 300 resolution. At the output, object detections (class, detection confidence, object coordinates) and their "deep features" are formed.

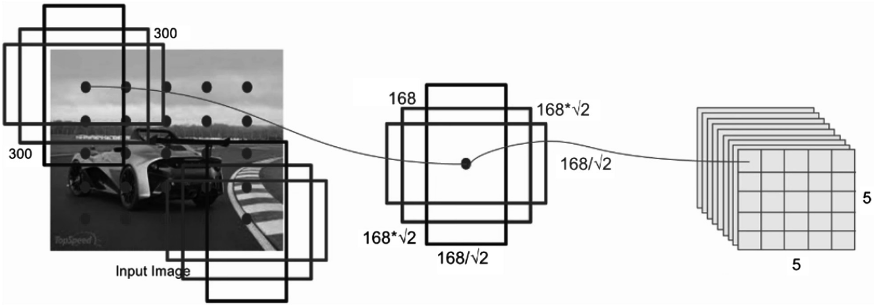

Fig. 3. Covering the original image with a grid of "anchor" boxes for a 5 × 5 feature map

Thus, applying the proposed model, it becomes possible to detect the image and get the "deep features" of detection in one pass of the CNN.

2.2 Algorithm for Obtaining "Deep Features" of the Objects Detection

Input Data: a feature map of the m-layer CNN $n \times h_m \times w_m$, coordinates of the object in the image $o = (x_1, y_1, x_2, y_2)$, image dimensions $h \times w$, dimension of the ROI-pooling layer $n \times h_r \times w_r$.

Output Data: a set of features P of the object o.

Step 1. Calculation of the scaling factors:

$$K_h = \frac{h_m}{h}, K_w = \frac{w_m}{w}.$$

Step 2. Calculation of the coordinates of object projections on a feature map:

$$x_i^m = x_i \cdot K_h, y_i^m = y_i \cdot K_w, i = 1, 2.$$

Step 3. Getting features of the object P_{proj} with dimension $n \times (y_2^m - y_1^m) \times (x_2^m - x_1^m)$ and coordinates $(x_1^m, y_1^m, x_2^m, y_2^m)$ on m-layer CNN feature map.

Step 4. Scaling of projections of the features P_{proj} object o on feature map with coordinates $(x_1^m, y_1^m, x_2^m, y_2^m)$ and dimension $n \times (y_2^m - y_1^m) \times (x_2^m - x_1^m)$ to dimension $n \times h_r \times w_r$ using max-pooling (ROI-pooling) $Ceil\left(\frac{y_2^m - y_1^m}{h_r}\right) \times Ceil\left(\frac{x_2^m - x_1^m}{w_r}\right)$, where *Ceil* – ceiling operator. As a result of this operation, a set of P features is formed. In the same way feature sets are computed for all the detections which are sent to the input of the ROI-Poling.

The distance between two sets of features P_1 and P_2 of two detections can be calculated as the norm of the difference between two vectors of dimension $n \times h_r \times w_r$ in Euclidean space:

$$d = \|P_1 - P_2\|.$$

The disadvantage of this approach is that additional normalization is necessary, so it is sometimes more convenient to calculate the distance between feature sets using the cosine distance:

$$d = \frac{P_1 \cdot P_2}{\|P_1\| \cdot \|P_2\|}, d = [0, 1]$$

3 Method of Dynamic Visual Scene Analysis

3.1 Formulation of the Problem

Let $K = \{k_n\}$ be a set of frames (a dynamic visual scene) n = 1, …, N, $k_n : O_{k_n} = \{o_{m_n}^{(k_n)}\}$ be a set of object in the frame k_n, $m_n = 1, \ldots, M_n$, n = 1, …, N.

Each of the objects $o_{m_n}^{(k_n)}$ is represented by a set of features $P_{m_n} \forall o_{m_n}^{(k_n)} : P_{m_n} = \left\{p_l^{(m_n)}\right\}, l = 1, \ldots, L$ and coordinates $\left\{\left(x_1^{(m_n)}, y_1^{(m_n)}\right), \left(x_2^{(m_n)}, y_2^{(m_n)}\right)\right\}$, $TR = \{tr_z\}$ is a set of tracks $z = 1, \ldots, Z$.

Each of the tracks tr_z is represented by a sequential representation of a single object $o_z^{(k_i)}$ in a sequence of frames k_i:

$tr_z = \left\{o_z^{(k_i)}\right\}, i = i_{in}, i_{in+1}, \ldots, i_{out}, z = 1, \ldots, Z, i_{in}, i_{out} \in K, i_{in} \leq i_{out}, o_z^{(k_i)} \in O_{k_i}$ moreover, the same object is not included in different tracks. Let θ be the degree of similarity between the sets, S be a method of dynamic visual scene analysis.

It is required to find $TR^* = \{tr_z^*\}$, a set of tracks, each of which is represented by a sequential representation of a single object $o_z^{*(k_i)}$ on a sequence of frames k_i, tracked using S methods, $z = 1, \ldots, Z$ such that

$$\theta(TR,\ TR^*) \xrightarrow{S} \text{max, with the set of constraints T.}$$

3.2 Description of the Method

The developed method differs from the existing ones in detection of objects and calculation of their “deep” features in one pass of the detector based on the CNN work in conditions of uncertainty or noise detection data.

This method allows one to select objects on each frame using a detector and compare them with existing tracks (sequences of object coordinates on previous frames) or delete a track if the sequence of frames associated with the track is missing.

Input Data: $K = \{k_n\}$, ρ_{thr} is a maximum allowed threshold of difference between a track and an object, $Count_{\max}$ is a maximum allowed number of consecutive frames, for which the degree of difference ρ between an object and a track is greater than ρ_{thr}, n_{av}

is the number of frames on the basis of which the averaging features P_{m_n} of the object $o_{m_n}^{(k_n)}$ with the coordinates $\left\{\left(x_1^{(m_n)}, y_1^{(m_n)}\right), \left(x_2^{(m_n)}, y_2^{(m_n)}\right)\right\}$ is performed.

Output Data: $TR = \{tr_z\}$ is a set of tracks, each of which is represented by the coordinates $\left\{\left(x_1^{(m_n)}, y_1^{(m_n)}\right), \left(x_2^{(m_n)}, y_2^{(m_n)}\right)\right\}$ of an individual object $o_z^{(k_n)}$ in a sequence of frames $k_n, o_z^{(k_n)} \in O_{k_n}$.

Initialization of a set of tracks, the number of which on the given frame initially equals zero.

Stage 1. The definition of the detector operating conditions is the uncertainty or noise in data detection.

If the operating conditions of the detector are unknown, any of the conditions can be selected, and then the correctness of the selection based on the detection accuracy indicators is evaluated. Such indicators can be represented as the number of false positives and the number of false negatives. If there are a lot of false positives of the detector, we can talk about noise detection data, otherwise the uncertainty of detection data is implied. The following sequence of stages is performed for each k_n:

Stage 2. A search of the coordinates $\left\{\left(x_1^{(m_n)}, y_1^{(m_n)}\right), \left(x_2^{(m_n)}, y_2^{(m_n)}\right)\right\}$ for each object $o_{m_n}^{(k_n)}$, the features P_{m_n} of this object using an appearance object model and an algorithm for obtaining "deep features" of the objects detection. This step may also include a filtering procedure in case of noisy detection data.

Thus, the stage is designed to search for the objects on the current frame and their features, which can then be assigned to the appropriate tracks on the basis of the comparison of the attributes of these objects.

Stage 3. Matching of the objects $\left\{o_{m_n}^{(k_n)}\right\}$ found in stage 1 with the existing tracks $\left\{tr_{z_{n-1}}\right\}$, if $TR \neq \varnothing$:

Step 1. Addition to the circular list $P_{m_{n-1}}$ and displacement $P_{m_{n-1-n_{av}}}$.

Step 2. Calculation of the average-object features P_z^C of the track tr_z as the arithmetic mean of the object features in the circular list:

$$P_z^C = \frac{\sum_{j=1}^{n_{av}} P_{m_{n-j}}}{n_{av}}.$$

At the same time, if the number of features in the list is less than n_{av}, the average-object features are calculated on the base of the existing number of features.

Step 3. Construction of the cost-matrix $m_n \times z_{n-1}$:

Each element of the matrix is the distance between features P_{m_n} (the total number of objects – m_n) and the features of the track average-object P_z^C (total number of tracks – z_{n-1}) based on the appearance model.

Step 4. Solution of the problem of assigning m_n objects to z_{n-1} tracks using the Hungarian algorithm [4].

As a result of the stage, some of the found objects will be assigned to the existing tracks.

Stage 4. Initialization of new tracks for objects from stage 3 that have not been assigned to tracks:

$$tr_{z_n} = tr_{z_{n-1}} \cup \left\{ o_{m_n}^{k_n} \notin tr_{z_n} \right\}.$$

When Step 3 is completed, the objects that could not be assigned to the tracks remain. This step is designed to create new tracks for such objects.

Stage 5. Check of the validity of detection assignment to a track. The assignment is considered valid in case the degree of difference between the track and the designated object does not exceed the allowable threshold $\rho\left(tr_{z_n}, o_z^{(k_n)}\right) \leq \rho_{thr}$.

The stage allows to exclude those detection assignments on the track, which are not allowable and represent false detector triggering.

Stage 6. Deletion of tracks from $\{tr_{z_n}\}$ if the number of frames in the sequence is greater than the threshold number of frames $k_{n-m}, \ldots, k_n, m \geq Count_{\max}$ at which the degree of difference between the track and the assigned object exceeds the allowed threshold $\rho\left(tr_{z_n}, o_z^{(k_n)}\right) \geq \rho_{thr}$.

The stage allows to delete those tracks on which the object was missing (perhaps, it was occluded by another object) several frames in a row.

Stage 7. Selection of tracks from $\{tr_{z_n}\} : \rho\left(tr_{z_n}, o_z^{(k_n)}\right) \leq \rho_{thr}$.

The stage allows one to select only those tracks on which the object is present. The tracks on which an object is overlapped by other objects or is missing are not displayed.

Thus, as a result of the stages of the developed method, a set of tracks $\{tr_{z_n}\}$ is formed on each frame k_n.

4 Effectiveness Evaluation of the Developed Method

Effectiveness evaluation of the developed method is carried out with the help of a training and test dataset 2D MOT15, so this technique corresponds to the recommendations given in [6]. The effectiveness comparison of the developed method with other existing methods (SORT, ELP, MHT, DPM) was carried out taking into account the following conditions: uncertainty of detection and noise detection data. To simulate the conditions of uncertainty, detection that was obtained using an appearance model of the object was taken. To simulate the noise conditions of the detection data, the detections from the 2D MOT 15 dataset was taken. Then the gained detection was combined into tracks using the method of dynamic visual scene analysis.

Furthermore, the proposed method was further implemented with the procedure of filtration in case of noise in the data: from the dataset 2DMOT15 in each frame the detection with detection confidence at least 30% was chosen. For each of such detection with the help of the algorithm for getting "deep features" of object detection, its "deep features" were formed. Then these features were compared in pairs and the detection with a difference degree less than 0.2 (selected experimentally) was selected. The detection, which has a lower detection confidence threshold, was excluded from each of such pair.

In addition, for the tracks that have a minimum frames number of successful track detections greater than 3, this detection confidence threshold was reduced to 20%.

The degree of difference ρ in this method between an object and a track was calculated on the basis of the cosine distance with the maximum difference threshold $\rho_{thr} = 0.5$ selected for the following reasons: the object can be overlapped by another object or changed between adjacent frames by no more than the half.

The remaining parameters are selected as follows: $n_{av} = 10$, $Count_{\max} = 3$.

Table 1 presents the obtained results. The table shows that the proposed method in case of noise detection data exceeds the methods that work online (SORT, DPM) and works online, but at the same time worse than the methods that work offline (ELP, MHT). With the uncertainty of detection, the proposed method is strongly superior to SORT.

Table 1. Evaluation results of the method effectiveness

Method	*MOTA*, % (noise detection data)	*MOTA*, % (uncertainty of detection data)	Online
SORT	26.0	20.2	Yes
ELP	30.3	28.1	No
MHT	34.2	In these conditions, the test fails	No
DPM	26.8	In these conditions, the test fails	Yes
Proposed method	27.3	26.3	Yes

The time spent on processing one frame of the proposed CNN was experimentally estimated using the appearance object model on GPU TITAN BLACK. These experiments do not take into account the time of reading the image from the hard disk, its transformation and loading in the CNN.

t_1 is one frame time processing using the proposed appearance object model;

t_2 is one frame time processing using SSD model detection without a ROI-Pooling layer;

t_3 – time for obtaining features on the frame based on CNN VGG 16.

The results of the comparison are presented in Table 2.

Table 2. Execution time comparison of different parts of the algorithm

Time, ms	t_1	t_2	t_3
	42	41	18

Thus, the use of the developed algorithm allows one to reduce frame processing time by 18 ms for each detection.

5 Conclusion

In this article we proposed a method of dynamic visual scene analysis based on the appearance model. This method allows one to detect images on the offered model and obtain "deep features" of detection for one CNN pass, to increase accuracy of tracks construction, and to execute processing in real time at the 24 FPS. In addition, this method can work both in conditions of uncertainty and in conditions of noise detection data.

Time spent on one frame processing using the appearance object model on GPU TITAN BLACK was experimentally estimated. The assessments results in conclusion that the application of the developed model allows one to reduce the processing time of the frame by 18 MS for each detection.

Acknowledgments. The work was supported by grant RFBR № 18-07-00928_a "Methods and technologies of intelligent support for research of complex hydro-mechanical processes in conditions of uncertainty on the convoluted neuro-fuzzy networks".

References

1. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. arXiv preprint arXiv: 1602.00763 (2016)
2. Garanin, O.I.: Method for selecting receptive field of convolutional neural network. Neurocomputers (3) 63–69 (2017)
3. Garanin, O.I.: Tuning method of multiscale model for detecting visual objects in a convolutional neural network. Neurocomputers (2) 50–56 (2018)
4. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**, 83–97 (1955)
5. Kim, C., Li., F.: Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4696–4704 (2015)
6. Leal-Taixe, L., Milan, A., Reid, I., Roth, S., Schindler, K.: MOTChallenge 2015: towards a benchmark for multi-target tracking. arXiv preprint arXiv: 1504.01942 (2015)
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E.: SSD: single shot multibox detector. arXiv preprint arXiv: 1512.02325 (2015)
8. McLaughlin, N., Rincon, J. M. D., Miller, P.: Enhancing linear programming with motion modeling for multi-target tracking. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 271–350 (2015)

9. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: learning to track multiple cues with long-term dependencies. arXiv preprint arXiv: 1701.01909 (2017)
10. Simonyan, K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556 (2015)
11. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. arXiv preprint arXiv: 1703.07402 (2017)