EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

Development of the video stream object detection algorithm (VSODA) with tracking

A.Y. Zarnitsyn[1,*], A.S. Volkov[1], A.A. Voycehovsky[1] and B.I. Pyakillya[1]

[1]Tomsk Polytechnic University, ayz10@tpu.ru, artvolkov96@yandex.ru, alexvoit70@gmail.com, pakillaboris@gmail.com

Abstract

The object tracking is one of the most important task in video analysis. Many methods have been proposed such as TLD (Tracking, Learning, Detection), Meanshift and MIL but they show good accuracy in laboratory cases, not in real ones, where the accuracy is defined as a numerical difference between computed object coordinates and the real ones. One of the reasons is lack of information about tracked object and environment changes. If a method has the prior information about tracked object, then it will be able to perform with higher accuracy. Some of the newest object tracking methods such as GOTURN use trained CNN (convolutional neural network) and have better accuracy because of knowledge about how the tracked object looks like in different situations such as light intensity changes and tracked object's rotations.

If we use only a classification algorithm (classifier) then it can find an object that was in training set with high probability. But if its appearance is changing it will be lost when deviation will be higher than trust limit. Then it is important to have parts of prior and posterior information about tracked object. The prior information is given by detector (CNN) and posterior information – by tracking algorithm (TLD). One of the biggest detector problems is high computational complexity in terms of operations' number and one of the solutions is to use the classifier in parallel with the tracker.

In future work we are going to use different sensors, not only RGB camera, but RGBD camera, which may improve accuracy due to higher amount of information.

Keywords: Computer vision, deep learning, machine learning, pattern recognition, mobile robotics, object tracking, video analysis.

Received on 19 April 2018, accepted on 10 November 2018, published on 28 January 2019

doi: 10.4108/eai.22-1-2019.156385

*Corresponding author. Email:ayz10@tpu.ru

1. Introduction

The video object tracking is one of the video analysis tasks. In case of object tracking we restrict an object area by means of bounding box in a first video frame. Then the algorithm finds object on next video frames. The existed tracking algorithms use only posterior information about object and it is their main disadvantage. The work's main point is to upgrade one of the best existed tracking algorithms by adding knowledge base about tracked object. The information about object will be obtained from a neural network, which will be trained to classify objects in video frames and will correct a tracker.

In our research we use the popular open-source computer vision library OpenCV, which has many important off-the-shelf functions e.g. very efficient tracking and classification algorithms [1].

2. Review of the methods

2.1. Tracking methods

There are two methods for object tracking: recognition and tracking. In case of recognition, the program knows how object looks and it sequentially checks parts of the image to find similar objects. Weakness of this approach is impossibility to find object, if it is overlapped – partially or fully. Also, it is possible if appearance of the object changes too much.

In case of tracking, the object is selected manually at the initial time, and then the program will track the object by estimating the optical flow. Optical flow characterizes relative change of objects locations on the next video frame. As the program knows where the object was in previous frames, it knows speed and motion direction of the object. This data allows to predict the next location of the object with high accuracy. In the case of a short-term loss of the tracking object from the lens' field of view, the program will no longer be able to track it.

An analysis of the above-mentioned shortcomings made it possible to work out a solution consisting in using a combined method.

At the initial time, a trained neural net detects object in the frame and sends coordinates of its bounding box (high left and bottom right corners) to the object tracking algorithm. After some time, the neural net detects object again to determine its new location. Thus, the work of the tracking algorithm is adjusted to avoid a loss of the object. Furthermore, our method makes it possible to save computing resources due to intermittent work of the neural network.

2.2. Tracking algorithms

Most of the tracking algorithms are implemented in the OpenCV – computer vision library, which has a large set of functions for working with images and video stream. For this research, we selected algorithms with maximum accuracy and speed of operation. The speed of operation is measured using processed number of frames per second (FPS). Below is a brief description of each of the algorithms, as well as their advantages and disadvantages.

I) MIL (Multiple Instance Learning).

MIL uses current location of the object as positive example for classifier learning. Several parts of the image that are equal in size to the object and are in a small neighborhood next to the current location of the object are used as potentially positive examples.

Thus, if the current location of the tracking object is not accurate, a positive example with exact current location may be in the set of potentially positive examples. In particular, given a training data set $\{(X_1, y_1), \dots, (X_n, y_n)\}$ in current frame, where a bag $X_i = \{x_{i1}, \dots, x_{im}\}$ and $y_i = \{0,1\}$ is its label, as well as a pool of M candidate weak classifiers $H = \{h_1, h_2, \dots, h_m\}$, MIL sequentially chooses K weak classifiers from the candidate pool based upon the following criterion

$$h_k = argmaxL(H_{k-1} + h) \quad (1)$$

where

$$L = \sum_i (y_i \cdot \log p_i + (1 - y_i) \cdot \log(1 - p_i)) \quad (2)$$

is the log-likelihood over bags, and $H_{k-1} = \sum_{i=1}^{k-1} h_i$ is the strong classifier consists of the first $k-1$ weak classifiers [1].

MIL works stable at partial overlapping of the object, but at short-term fully overlap the algorithm loses object of tracking.

II) TLD (Tracking, Learning, Detection).

As the name suggests, this algorithm breaks long-term tracking into three components: short-term tracking, learning, and recognition [2]. First component tracks the object frame-by-frame. Second component corrects the tracking module, if necessary. Third component is learning module. It accumulates images of the object and seeks to reduce the error. The search space is restricted to

$$numWindows = \sum_{s \in 1.2^a} \left[\frac{n - s \cdot (\omega + d_x)}{s \cdot d_x} \cdot \frac{m - s \cdot (h + d_y)}{s \cdot d_y} \right] \quad (3)$$

Where $s \in 1.2^a$, $a \in -maxScale \dots minScale$, n is image width, m is image height, ω is width of initial bounding box, h is height of initial bounding box, d_x and d_y are margins between two adjacent subwindows and set to be $\frac{1}{10}$ of the values of original bounding box [2].

The scheme of the algorithm is shown in Figure 1.

Among the advantages it is worth to notice that the work of the algorithm goes in real time, the stability of long-term overlaps of the tracking object and the stable tracking in case of object scale changing.

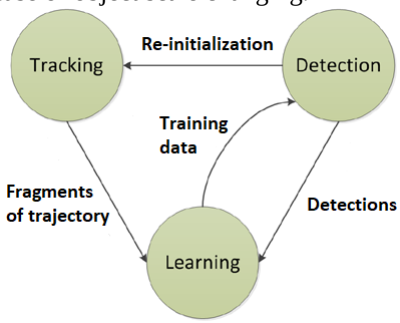

Figure 1. Scheme of the algorithm TLD

III) GOTURN.

GOTURN is based on convolutional neural network (CNN), which has already been trained on the large dataset of various objects. CNN seeks to find object selected at the initial time on each frame. The tracking object must belong to the classes of objects that the CNN is capable of recognizing.

The general principle of work is shown in Figure 2.

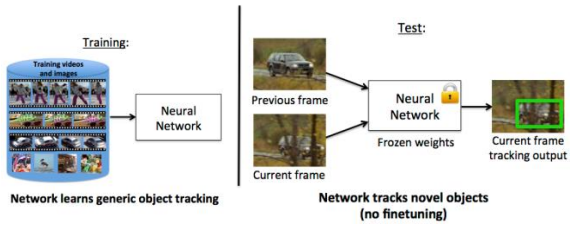

Figure 2. Scheme of the algorithm TLD

Advantages of this algorithm include the correctness of determining the loss of an object. The disadvantages are the low stability of tracking in cases of overlapping of the object.

2.3. Neural nets for classification

Convolutional neural networks are usually used in image and video recognition, sometimes in natural language processing etc. This kind of neural net architecture has been suggested by Canadian scientist Yann LeCun [3].

The principle scheme of the typical deep CNN (DCNN) is shown on Figure 3.

The advantages of convolutional neural networks:

- CNNs have less set of customizable weights than fully connected networks with equal accuracy.
- The possibility of using parallel computation, including computation on GPUs, which speeds up the learning and operation of the neural network.
- Resistance to image distortion, such as shift.
- High accuracy of image classification [9].

Main disadvantage of CNNs is high requirements for computational resources. If there is enough CPU to classify images using CNN, then powerful GPUs are required to solve the detection task.

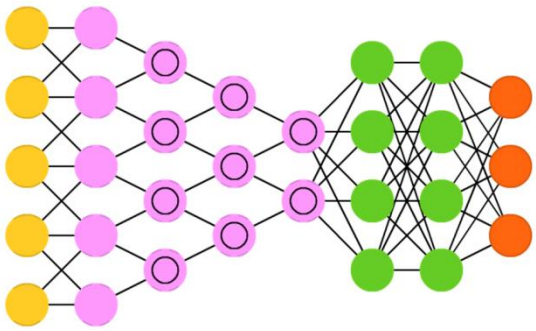

Figure 3. Scheme of the typical DCNN

2.4. Object detection algorithm

As a detection algorithm to analyse, we chose YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories [4] On a GPU Titan X it processes images at 40-90 FPS and has a mAP on VOC 2007 of 78.6% and a mAP of 48.1% on COCO test-dev.

The algorithm applies a single convolutional neural network to the full image and the network divides the image into regions and predicts the bounding boxes and probabilities for each region. These bounding boxes are weighed according to the predicted probabilities.

YOLOv2 as an update of YOLO9000 has the best performance and detecting accuracy ratio [5].

Despite its advantages YOLO has one serious problem – very high requirements for computational resources. It makes impossible to use YOLO directly on mobile platforms such as Raspberry Pi etc.

3. Materials and methods

3.1 Algorithm description

Basing on existed tracking algorithms TLD is chosen as basic algorithm. Convolutional neural network (CNN) is used for object detection. We selected this type of neural networks due to the advantages listed in the literature review.

We decided to use a state-of-the-art detector YOLO 9000. This detector based on Darknet – neural network framework which has been developed with C and CUDA [6].

We use Darknet's realization on Tensorflow framework – Darkflow with weights trained on Imagenet dataset [7][9].

Input of detector takes an image with three color channels (RGB). Output of detector return five parameters: object class, top left corner coordinates, width and height of bounding box.

In the beginning it is planned to use own NN model with detector, but its accuracy was worth than state-of-the-art detectors.

The algorithm has next work principle. The initial video frame is processed by CNN, which classifies objects and determines its position on the frame. Then the information is transferred to TLD tracker and it tracks the necessary object. Periodically the object position from classifier and tracker is comprised. If difference is bigger than the threshold TLD will get new tracked object position which is determined by CNN.

CNN plays the expert role and its information about the tracked object is more important than from TLD. In situations when CNN cannot find the tracked object the

information from TLD is used. The main program algorithm is presented on figure 1.

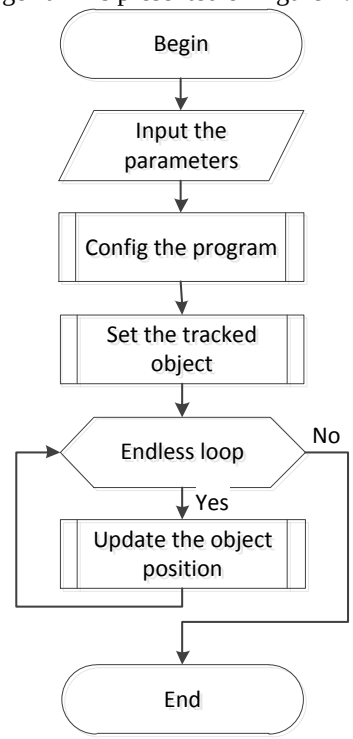

Figure 1. Main program algorithm

4. Results

Addition of the CNN to the tracking algorithm solves the problem of object loss in case of its disappearance from the frame. To verify that CNN usage increases the tracking quality we checked algorithm work with and without CNN on test videos. The tracking quality is ratio between time when the algorithm works correct and video length. The results are shown in table 1.

Table 1. Comparison tracking quality with and without CNN

Video name	TA	TA+CNN	Difference, %
Case 1, Video 1	0.40	0.52	+30.0
Case 1, Video 2	0.26	0.77	+196.15
Case 2, Video 1	0.20	0.28	+40.0
Case 2, Video 2	0.22	0.24	+9
Case 3	0.98	0.99	+1

The first column presents list of test videos. Case 1 demonstrates the partially overlapped tracked object. Case 2 shows the fully overlapped tracked object. In case 3 the distance between the tracked object and the camera is changing.

The second column (TA) – usage only tracking algorithm. The third column (TA+CNN) – combination of tracking algorithm and CNN (our algorithm). The fourth column is relative difference between TA+CNN and TA methods.

When the background is static and the tracked object is fully presented on the video frame, the simple tracking algorithm can be used. But in more difficult situations CNN helps to find the tracked object when it is partially overlapped.

The program is developed for two operating systems (OS): Windows and Linux. We tested our program on the Windows-laptop with CPU Intel Core i5-7200U, 8 GB DDR4 RAM and GPU NVidia GeForce GTX 950M. The initialization of the detector takes about 8 seconds. The tracker operates at a rate of 5-12 FPS. The detector processes one frame in 3-4 seconds. This allows robot to track static or not very fast objects. The detector tuned to recognize two classes of objects – cups and sport balls.

We also tried to test our algorithm on Raspberry Pi 3 with Raspbian OS, however we failed due to lack of RAM for the detector YOLO.

Examples of the work of our program is shown on the figures 4-7. Red bounding boxes are provided by the tracker, green bounding boxes – by the detector.

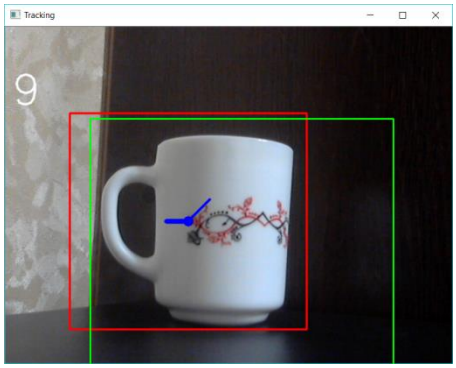

Figure 4. Object is not overlapped

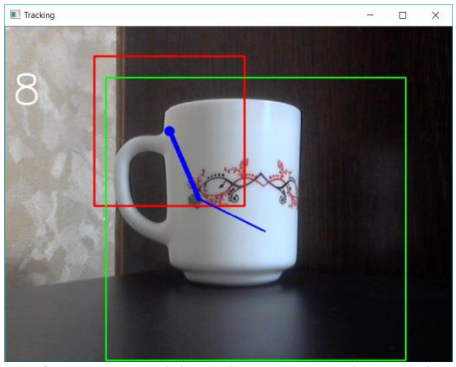

Figure 5. Object is not overlapped

When the object is not overlapped the tracker works correctly, the detector finds object (figures 4 and 5). Then the detector returns bounding boxes larger than object.

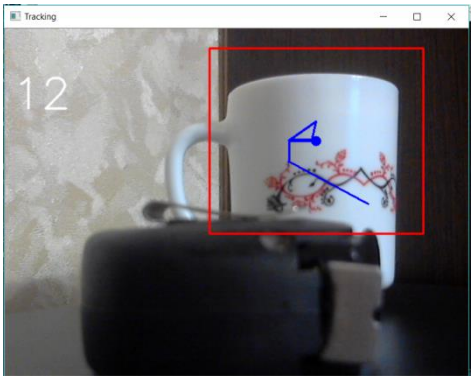

Figure 6. Object is overlapped partially

When the object is overlapped partially by the black block, the tracker often continues to track, but the detector sometimes cannot find the object. This is due to specific tuning of the detector YOLO for recognition many classes of objects. It recognizes black block as TV or monitor with high probability, which greater than probability of the target object. We plan to solve this problem in the future through additional tuning of the detector.

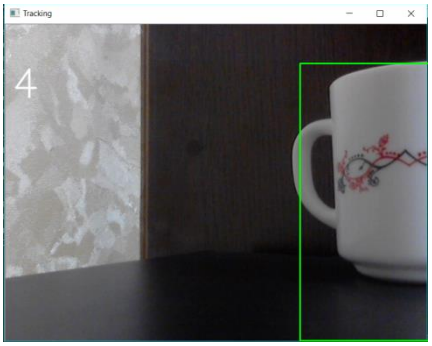

Figure 7. Object partially out of frame

When the object partially out of frame the tracker loses it, but the detector can find the object. Thus, our algorithm solves problem of redefinition the target object in case the object disappeared from the frame and came back.

In the future, it is planned to scale the platform, as well as increase the performance and quality of the algorithm due to using NVidia Jetson.

5. Conclusions

During this work, an algorithm was developed to track objects of two classes. The program can be used to control mobile robot using relative object position on the video frame and it can obtain information from different sensors such as range sensors or motor encoders and uses it to compute control signal for the robot.

Acknowledgements.

The research is carried out at Tomsk Polytechnic University within the framework of Tomsk Polytechnic University Competitiveness Enhancement Program.

References

[1] E.V. Bulatnikov, A.A. Goeva “Comparing computer vision libraries for using in application that uses the recognition technology of flat images”, *Magazine Vestnik MGUP by Ivan Fedorov*, vol. 6, pp 85-91, 2015.

[2] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, “Tracking Learning Detection”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 6, no. 1, 2010.

[3] Yann LeCun, Yoshua Bengio “Convolution Networks for Images, Speech, and Time-Series”, *The handbook of brain theory and neural networks*, pp 255-258.

[4] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 6, no. 1, 2010.

[5] “YOLO: Real-Time Object Detection” [Online]. Available: https://pjreddie.com/darknet/yolo/

[6] “DarkNet Github” [Online]. Available: https://github.com/pjreddie/darknet

[7] “DarkFlow Github” [Online]. Available: https://github.com/thtrieu/darkflow

[8] Boris Babenko, Ming-Hsuan Yang, Serge Belongie, "Robust Object Tracking with Online Multiple Instance Learning", IEEE TPAMI, 2011

[9] “ImageNet” [Online]. Available: http://www.image-net.org/