Towards More Efficient Deep Tracking

Zan Huang
August 3rd, 2017

***Abstract*—Visual object tracking is a hot topic for computer vision research in recent years. Many visual object tracking algorithms based on deep learning technique appeared, they achieved state-of-art performance on existing benchmarks. These algorithms commonly adopt deep neural networks for feature extraction. Most of these trackers are generations of feeding large volume datasets to artificial neural networks. The effectiveness of deep features for visual tracking has been shown in benchmarks, these big guys showed more intelligent behavior capturing target object in bounding box but the high computational cost and the deep hunger for data contradict our initial requirements for practical tracking algorithm. In this project, we focus on single-object tracking algorithm to seek for proper configuration for constructing more efficient deep trackers.**

***Index Terms*—Visual object tracking, Deep learning.**

I. Domain Background

VISUAL object tracking algorithm[2] is essential for many computer vision applications. Trackers serve in a wide range of systems including video surveillance, self-driving cars, and human-computer interaction. For single-object tracking, trackers need locating the object of interest marked in initial video frame continuously. Given a bounding box, best tracker needs to overcome challenges like fast motion, serious occlusion, object deformation and achieve real-time performance with high precision. The inequity of requesting better performance and generality in seriously constrained context with quite few information made it a challenging computer vision task. To win the challenge, many tracking algorithms have been proposed including ones with powerful deep learning engines over the past decade. There is a trend to exploit automatically extracted deep features for visual tracking task and construct an end-to-end system to track. While high computational cost and execution speed hold trackers back from practical use, problems like overfitting also perplex these approaches. The emerging technology revolution empowered by artificial intelligence related research has brought us closer to the implementation of systems only seen in science-fictiony movies before. Artificial vision systems built in complex modern artifacts are very important for supporting their execution and perception based on camera captured raw pixels is one of the keys to open the machine intelligence.

II. Problem Statement

In this paper, we focus on single-object tracking, the context of this task could be described as taking a video sequence and a bounding box localizing the target object in the initial frame as input, implement an algorithm to generate consecutive bounding boxes as output.

This file was submitted as capstone proposal for machine learning Nanodegree at Udacity. Revised on April 18, 2018. Meanwhile, I need to mention that all *we* just refer to myself but as a habit I pick this work to represent the authorship.

III. Datasets and Inputs

OTB: http://cvlab.hanyang.ac.kr/tracker_benchmark/ will be the dataset used for tracker performance evaluation in our experiment. Other benchmarks like VOT: http://votchallenge.net/, PTB: http://tracking.cs.princeton.edu/ will also be considered if necessary. These benchmarks typically consist of three parts: video data, human marked ground truth, and evaluation metrics, but just for the dataset aspect, we only discuss video and ground-truth data at this section.

Formally, the input for tracking algorithms are video frames I and initial bounding box B_0. Conventionally, a bounding box may consists of four numbers:

1) x: x-value for upper left corner of B
2) y: y-value for upper left corner of B
3) w: width of B
4) h: heigh of B

Take OTB as an example, the full dataset(OTB 100) consists of 98 video sequences and 100 objects were labeled by bounding boxes. Each video sequence could be downloaded as a separate zip file. After extracting, each video sequences as a folder has corresponding frame images(video resolution is arbitary and both gray/RGB images are included,) inside ordered by their time stamps. Meanwhile, each sequences will have one or two ground-truth text files, in which each line corresponds the the bounding box parameters of the target at certain time stamp. More details are accessible at corresponding website which we strongly suggest the readers to checkout.

For historical reasons, most tracking benchmark only serve as testing set. In reference to SiameseFC which serves as the benchmark in this paper, we will use IMAGENET-VID-2015 data for training, the naming convention and dataset is a more complex and we suggest reading official documentation(http://image-net.org/challenges/LSVRC/2015/#vid) for reference, in short, VID will be RGB video frames for different sequences, arbitrary resolution for each sequence, one or more objected marked by bounding box in each frame and saved to xml files. One thing that worth noting is, we will sure use the same datasets for training, validation and testing as in SiameseFC[1] which was picked as benchmark in our experiment.

IV. Solution Statement

In general, the aim of our tracking algorithm is to accurately and efficiently output bounding box B_i for consecutive frame I_i to locate single target object marked in the initial frame.

CNN will be used for feature extraction and both fully connected layers and correlation filters will be considered for building up the complete tracking algorithm. The learning paradigm will be a combination of siamese method which uses the initial frame as an reference and a regression method which utilize recenty one or k frames to try to predict the bounding box parameters directly.

Visual attention methods will also be considered in algorithm development to reduce the noise coming from the environment and stay focused on the target. Recurrent neural network may not be used as they have trouble adapting to visual tracking task shown in few unpublished technical reports.

As a summary, we aim at using CNN, correlation filters, fully connected layers and visual attention methods to build up an efficient single object visual tracking algorithm enpowered by deep learning techniques. The real difficulty lies in distinguishing different objects and background by motion information and track the target against various annoying factors like illumination change, abrupt motion and occlusion, rather than applying fancy techniques to this specific task.

V. Benchmark Model

For the standard benchmarks, generated bounding boxes are compared with groundtruth which is absolutely computable and comparable.

We choose SiameseFC[1] as the benchmark model used in this experiment, the work has been published and is quite popular in recent years, for details of the algorithm, please refer to the original paper which is freely avaiable at https://www.robots.ox.ac.uk/~luca/siamese-fc.html. In short, this benchmark employs an alexnet similar CNN architecture to extract features from search region and target object instance, utilized correlation filtering to generate a fixed size response map to help localizing the object in the search region with reference to peak on the response map. This method used ImageNet VID 2015 dataset for training and rely on a cosine window to work in the tracking procedure. It has been tested on several tracking benchmarks including OTB and could run in real time with support of high-end GPU. And of course, we will keep using the same training set and testing set(OTB) in our own algorithm development.

VI. Evaluation Metrics

The overlap ratio between groundtruth and generated bounding boxes $\frac{S_{gt} \cap S_{gen}}{S_{gt} \cup S_{gen}}$ is main evaluation reference for the experiment, and for video sequences, average overlap rate is always employed to represent the tracking performance of certain tracker on one or more sequences. The larger the overlap ratio is, the better the tracker would be. Besides, different benchmarks(datasets) may have more detailed evaluation metrics, but the overlap ratio is always the common and most important one, for more evalution metrics explaniations, please check official sites like the one for OTB: http://cvlab.hanyang.ac.kr/trackerbenchmark/.

VII. Project Design

In general, we plan to investigate two current works for reference, namely, GOTURN[3] and MDNet[4]. Specifically, we will extract features by light-weight convolutional neural network and explore a better tracking algorithm by trail-and- error test. The whole system will be built on tensorflow and opencv, then evaluated on python version of OTB: https://github.com/jwlim/tracker_benchmark.

For the theoretical workflow, please check the pseudocode in this section. As the experiment is in progress, further modification will be introduced into the algorithm, detailed illustration and description will be given in final report.

Proposed Tracking Algorithm

procedure TRACKER(V, $bbox$)
    $instance = V\{1, bbox\}$
    for all $frame \in V\{2, ..., n\}$ **do**
        crop search region,
        extract feature,
        correlation filtering and regression,
        localization
    end for
    output results as bbox sequence.
end procedure

For specific stragies, we may employs hard negative mining or similar approach to abate the imblance of training data, and employ the ensemble method for combing the power of different models and approches to improve the tracking performance. No special pre-processing technique will be employed, we may just normalize the input image.

Here is will actually be done at each level. Tracking as a computer vision task do not focus on data due to historical reasons, researchers was squashing different sequence with different objects and challenging factors into a single benchmark, so sorry I could not give too much discussion on *data* temporally, but planned to add some in final paper rather in the proposal.

1) Preprocessing, crop instance in initial frame of each video sequence, do some tasks like image normalization if necessary;
2) Tracker construction, use CNN combined with fully-connected layers and correlation filter to perform localization task;
3) Init the weights of the network and train the tracker using backpropogation algorithm according to a specially designed loss function;
4) Validating and tuning parameters, testing on OTB.

References

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865. Springer, 2016.

[2] Z. Chi, H. Li, H. Lu, and M. Yang. Dual deep network for visual tracking. *IEEE Transaction on Image Processing*, 26:2005–2015, 2017.

[3] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference Computer Vision (ECCV)*, 2016.

[4] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *arXiv preprint arXiv:1510.07945*, 2015.