Deep Learning of Appearance Models for Online Object Tracking

Mengyao Zhai[1], Lei Chen[1], Greg Mori[1], Mehrsan Javan Roshtkhari[2]

[1]Simon Fraser University, [2]SPORTLOGiQ

Abstract. This paper introduces a deep learning based approach for vision based single target tracking. We address this problem by proposing a network architecture which takes the input video frames and directly computes the tracking score for any candidate target location by estimating the probability distributions of the positive and negative examples. An online fine-tuning step is carried out at every frame to learn the appearance of the target. The tracker has been tested on the standard tracking benchmark and the results indicate that the proposed solution achieves state-of-the-art tracking results.

1 Introduction

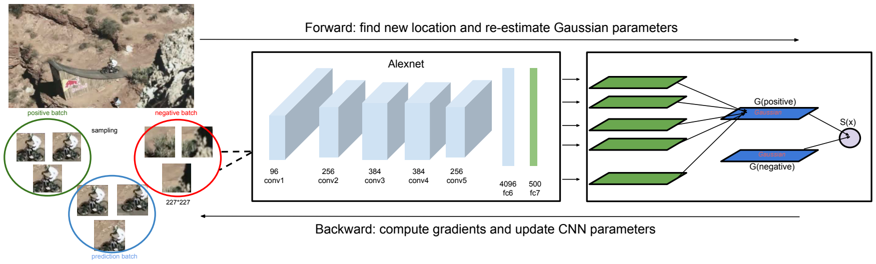


Fig. 1: Overview of our approach: Given one frame, we sample three batches: positive batch, negative batch, and prediction batch. In the forward procedure, given CNN parameters, we use the positive batch and negative batch to re-estimate Gaussian parameters. Then we search in the prediction batch for the new location with maximum score. In the backward procedure, given Gaussian parameters, we compute gradients with respect to feature nodes and update CNN parameters.

Visual target tracking is a fundamental task in computer vision and vision based analysis. In general, single target tracking algorithms consider a bounding box around the object in the first frame and automatically track the trajectory of the object over the subsequent frames. Readers may refer to [13] and [12] for a review of the state-of-the-art in object tracking and a detailed analysis and comparison of representative methods.

In this paper, we propose a new deep learning based tracking architecture (Fig. 1 shows the overall architecture of the proposed tracking system) that can effectively track a target given a single observation. The main contribution of this paper is a unified deep network architecture for object tracking in which the probability distributions of the observations are learnt and the target is identified using a set of weak classifiers (Bayesian classifiers) which are considered as one of the hidden layers. In addition, we fine-tune the CNN tracking system to adaptively learn the appearance of the target in successive frames. Experimental results indicate the effectiveness of the proposed tracking system.

2 Proposed Approach

This section presents the algorithmic description and the network architecture for the proposed tracking system. The system consists of a two stage training process, an *offline fine-tuning* procedure and an *online target specific fine-tuning* step.

Offline fine-tuning The fine-tuning of the pre-trained network is carried out through two phases: *obj-general* as phase 1 and *obj-specific* as phase 2. The first step is carried out by taking a pre-trained CNN which is already trained for large-scale image classification tasks, and then is fine-tuned for the generic object detection task which is referred to as *objectness* [1]. In order to learn generic features for objects and be able to distinguish objects from the background, we sampled $100k$ auxiliary image patches from the ImageNet 2014 detection dataset[1]. For each annotated bounding box, we randomly generate negative examples from the images in such a way that they have low intersection of union with the annotated bounding box. During this phase, all CNN layers are fine-tuned. The fine-tuned CNN can now be considered as a generic feature descriptor of objects, but it still cannot be used for tracking because it cannot discriminate a specific target from other objects in the scene. In other words, this network is equally activated for any object in the scene.

Another phase of fine-tuning is conducted given the bounding box around the target in the first frame. In order to generate a sufficient number of samples to fine-tune the network, we randomly sample bounding boxes around the original one. Those bounding boxes have to have a very high overlap ratio with the original bounding box. For the negative bounding boxes we sampled bounding boxes whose centers are far from the original one. During this phase, only fully connected layers are fine-tuned.

online target specific fine-tuning When a new frame comes, our model would take the features from the network and compute scores for all candidate bounding boxes as described below. Given a single bounding box representing the target of interest in the current frame of a video sequence (which can be initialized by either running an object detector or using manual labeling), first we use a sampling scheme to sample some positive patches around it and some negative patches whose centers are far from positive ones. Then, the probability density functions of the positive and negative examples are computed using (1). This process is repeated when a new frame comes.

[1] http://image-net.org/challenges/LSVRC/2014/

Similar to [2, 14], we assume that the distributions of the positive and negative examples' features can be represented by Gaussian distributions. Therefore, the posterior probability of the positive examples $P(\mathbf{x}|pos)$ is:

$$\begin{aligned} \mathcal{G}_{pos} &= P(\mathbf{x}|pos) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_{pos_i}} e^{-\frac{(x_i-\mu_{pos_i})^2}{2\sigma_{pos_i}^2}} \end{aligned} \tag{1}$$

where μ_{pos_i} and σ_{pos_i} are the mean and variance of the Gaussian distribution of the i^{th} attribute of the positive feature vector, x_i, respectively. Similarly, we can get distribution $\mathcal{G}_{neg}$ for negative examples.

Then the tracking score $\mathcal{S}(\mathbf{x})$ given an observation $\mathbf{x}$ is computed as:

$$\mathcal{S}(\mathbf{x}_i) = log\left(\prod_{i=1}^{n} \frac{P(x_i|pos)}{P(x_i|neg)}\right) = \log(\mathcal{G}_{pos}(\mathbf{x}_i)) - \log(\mathcal{G}_{neg}(\mathbf{x}_i)) \tag{2}$$

The candidate bounding box which has the highest tracking score is then taken to be the new *true* location of the target:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i) \tag{3}$$

Once the *true* target bounding box is determined in the following frame, the whole model shall be fine-tuned again in order to adapt itself to the new target appearance. We consider updating Gaussian parameters first, and then updating the network weights.

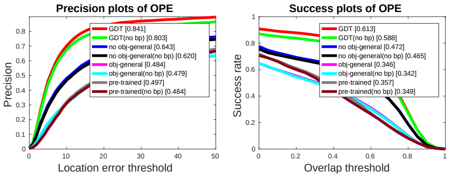


Fig. 2: Ablation Study

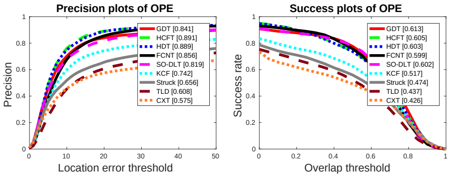


Fig. 3: Comparision with State-of-the-arts

3 Experiments

In order to evaluate the performance of our deep learning based tracker, we have carried out extensive experiments using the CVPR13 "Visual Tracker Benchmark" dataset [12]. We follow the "Visual Tracker Benchmark" protocol introduced in [12] in order to compare the tracking accuracy to the state-of-the-art approaches.

In our experiments, Opencv[2] and Caffe[3] libraries are used for the CNN-based tracking system. The CNN is fine-tuned for $100k$ iterations for objectness and the maximum number of iterations for the specific target fine-tuning in the first frame is set to be equal to 500. During online tracking, the CNN is backpropogated 1 iteration per frame. The aspect ratio is fixed as the same as the initialization given in the first frame of each sequence. The learning rate for Gaussian parameters is set to 0.95. The current prototype of the proposed algorithm runs at approximately 1 fps on a PC with an Intel i7-4790 CPU and a Nvidia Titan X GPU.

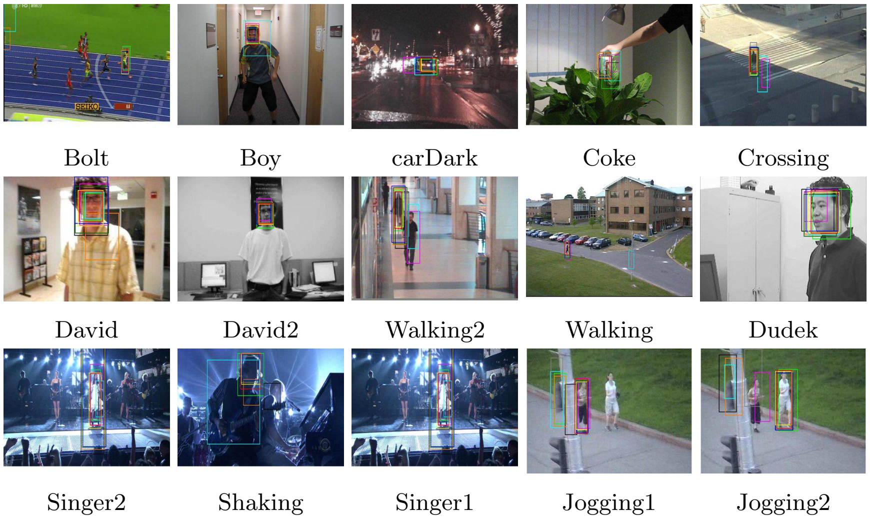
Ablation study For ablation study, we have conducted multiple experiments with three pairs of baselines. The first pair of baseline, which we refer to it as the ”pre-trained” is to take the pre-trained model [7] as the feature extractor (without fine-tuning for objectness and target appearance) and use the same tracker as GDT to track every target in each sequence. By ”no bp” we mean that during tracking process only Gaussian parameters are updated and CNNs are not fine-tuned. The second pair of baselines, which we call them the ”obj-general”, is to take the CNN model we trained for objectness as the feature extractor. To show the importance of fine-tuning for objectness, we add third pair of baselines, which we refer to as the ”no obj-general”. For this baseline, we remove the objectness step and CNNs are fine-tuned directly from the pre-trained model. All results listed in this section adopt same tracker, the only difference is the CNN models that are used. We summarize comparisons with all baselines in Fig. 2. From Fig. 2, it is clear that each step of our algorithm boosts the tracking results.

Comparison with state-of-the-art Our tracking results are quantitatively compared with the eight state state-of-the-art tracking algorithms with the same initial location of the target. These algorithms are tracking-by-detection (TLD) [6], context tracker (CXT) [3], Struck [4], kernelized correlation filters (KCF) [5], structured output deep learning tracker (SO-DLT) [11], fully convolutional network based tracker (FCNT) [10], hierarchical convolutional features for visual tracking (HCFT) [8], and hedged deep tracking (HDT) [9]. The first four algorithms are among the best trackers in the literature which use handcrafted features, and the last four are among best approaches for CNN-based tracking. **GDT** represents our proposed approach.

Fig. 3 shows the success and precision plots for the whole 50 videos in the dataset. Overall, the proposed tracking algorithm performs favorably against the other state-of-the-art algorithms on all tested sequences. It outperforms all of the state-of-the-art approaches given success plot and produces favourable results compared to other deep learning-based trackers given precision plot, specifically for low location error threshold values. We show some visualizations of detection results of all approaches in Figure 4.

[2] http://opencv.org

[3] http://caffe.berkeleyvision.org



Bolt Boy carDark Coke Crossing

David David2 Walking2 Walking Dudek

Singer2 Shaking Singer1 Jogging1 Jogging2

Fig. 4: Visualizations of all tracking algorithms on challenging sequences. Ground Truth: red, GDT(ours): yellow, FCNT: gray, HDT: dark green, HCFT: blue, SO-DLT: green, KCF: black, Struck: orange, TLD: magenta, CXT: cyan.

4 Conclusion

We proposed a novel tracking algorithm in this paper. The CNN for tracking is trained in a simple but very effective way and the CNN provides good features for object tracking. First stage fine-tuning using auxiliary data largely alleviates the problem of a lack of labelled training instances. A second stage of fine-tuning, though used only with a few hundred instances and trained for tens of iterations, greatly boosts the performance of the tracker. On top of CNN features, a classifier is learnt. The experimental results show that our deep, appearance model learning tracker produces results comparable to state-of-the-art approaches and can generate accurate tracking results.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 73–80. IEEE (2010)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(8), 1619–1632 (2011)
3. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 1177–1184. IEEE (2011)
4. Hare, S., Saffari, A., Torr, P.: Struck: Structured output tracking with kernels. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 263–270 (Nov 2011). https://doi.org/10.1109/ICCV.2011.6126251

5. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. Pattern Analysis and Machine Intelligence, IEEE Transactions on **37**(3), 583–596 (2015)
6. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 49–56. IEEE (2010)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
8. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3074–3082 (2015)
9. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged deep tracking (2016)
10. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3119–3127 (2015)
11. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arXiv:1501.04587 (2015)
12. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
13. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing **74**(18), 3823–3831 (2011)
14. Zhang, K., Zhang, L., Yang, M.H.: Real-time object tracking via online discriminative feature selection. Image Processing, IEEE Transactions on **22**(12), 4664–4677 (2013)