An Appraisal Theme for Deep Position Specific Tracing

Peddinti Siva Srinath [1], G. Madhusudhana Rao [2], P.Jayababu[3]
M.Tech scholar, Nannapananeni Venkatrao College of Engineering & Technology, Tenali
ECE HOD, Nannapananeni Venkatrao College of Engineering & Technology, Tenali
Asst. Prof., Nannapananeni Venkatrao College of Engineering & Technology, Tenal

Abstract- **In this paper an appraisal theme for visual tracing by using CNN. Due to various uncertain changes of articles online, such as sudden motion, background position and large deformation, the visual tracing is still a stimulating task. Hear a novel Theme for Deep Position-Specific Tracing. It differ the different criteria like quantization task and a grouping task, and in orders an individual network for each task. The quantization network achievements the data in the current edge and provides a selected position to improve the probability of successful tracing, while the grouping network finds the target in the middle of many examples generated around the target position in the previous edge, as well as the one predictable from the quantization network in the current edge. CNN based tracers often have substantial number of usable parameters, and are prone to over-fitting to some particular position states, leading to less precision or tracing error.**

Index Terms- **Visual Tracing; position Specific Tracing; CNN; Decomposition Network.**

I. INTRODUCTION

Visual position tracing is a core problem in multimedia understanding and computer vision. It has numerous applications in robotics, human-computer interaction, video analysis. In this paper, we focus on the problem of single position tracing. A typical system is to track an uninformed position in a video in order, where the position has been represented in the first edge by a rectangle. Although many progress has been made in recent years, it is still a challenging task due to unknown changes of position online, such as shape deformations, occlusions, fast motion and pose variations, to name a few. Many methods have been proposed for the visual tracing problem, such as multiple instance learning, subspace learnin, ensemble learning, compressive coding, etc. In recent years, the Discriminative Relationship Filters (DRF) based methods have achieved good results in terms of accuracy and speeds. More recently, several pure CNN based methods have been developed and obtained state-of-the-art tracing results in public benchmarks. Despite achieving promising performances, existing CNN trackers still have some drawbacks. First, to predict the position of target in the current edge, most of these trackers search the position near its position in the previous edge, which are prone to drifting in cases of fast motion and error locating in the previous edge. One way to invoke this problem is to sample as many examples as possible in a larger region and feed them into a network, resulting in very slow tracing speeds since CNN should run many times on the generated examples . Second, we argue that the convention state transition strategy for visual tracing is suboptimal because they do not exploit useful information from the current edge. Third, because only limited number of positive data can be used online, many top CNN trackers have large guideded parameters, and thus are likely to be over-fitting to backgrounds. To improve tracing accuracy and robustness, it is thus imperative to provide a tracker which can handle all above questions. In this paper, we propose a simple, flexible yet effective method, Deep position-Specific Tracing (DPST), which decomposes the single tracing task into quantization and classification, and trains different network for each task. Our architecture is significant difference from tracker combination methods because we train our networks for separated purposes: one network is for target position estimation, and the other is for frontend and background classification. Specifically, the quantization network is a small network operated on a searching region which is larger than the target size, and provides a specific guess of the target position in the current edge. The classification network accepts many examples generated from the target position in the previous edge and the specific one assured in the current edge. Finally, the extensive experimental results show that the proposed edgework achieves competitive performance compared with other state-of-the-arts on popular benchmarks.

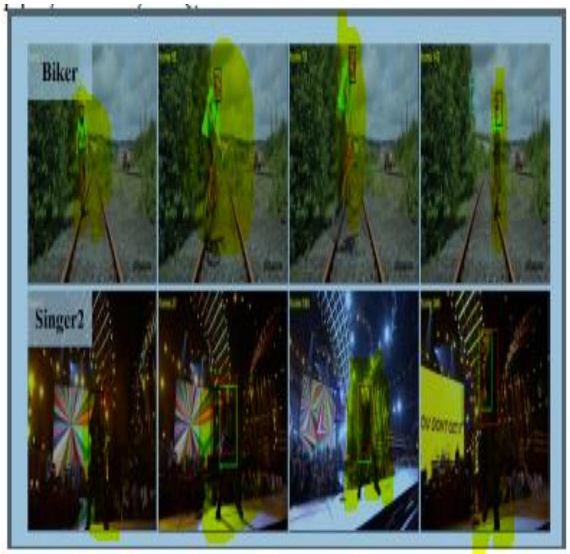
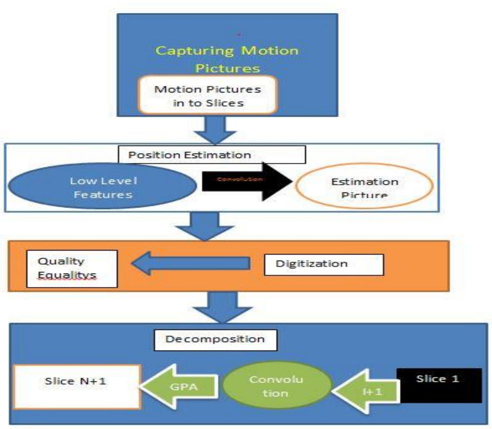


Figure: A comparison of the proposed DPST with MDNet and C-COT on two example sequences. The AUC on "Biker"/"Singer2" for DPST, MDNet and C-COT are 0.645/0.714, 0.392/0.694 and 0.531/0.081, respectively. Best viewed in color. Helighted The Position what we are going to Trace in a motioned picture.

2. DPST ALGORITHM



3. QUANTIZATION

The search region is 4 times larger than the target size, and it is resized to 107×107×31 as input to the pre-trained VGG-M [6]. Here, we adopt the low-level features from the VGG-M network (first ReLU layer) since it preserves more spatial information.

Snorm on estimated location. The proposed state transition strategy is denoted as

$S_{comb} = \{S_{norm}(X_t\ ; e_{xt-1});\ S_{norm}(X_t\ ; x_{*t}\ )\}$

The output is a 51 × 51 × 96 feature map which is then fed into our network for localization. The architecture of our network is implemented with two 1 × 1 convolutional layers, followed by a loss layer. The first convolutional layer has convolutional kernels of channel 96 and outputs 256 feature maps, followed by a non-linear ReLU unit and a dropout layer. The main goal of this layer is to adapt the pre-trained VGG-M features to specific videos since the same kind of objects can be treated as target or as background in different sequences. The second layer has kernels of channel 256 and outputs the heat map of the cropped region. To train the network, we employ softmax loss2 and define labels on the feature map as:

$y_i =$ 2, if $||c_t - e_{ct}|| \leq R$, Eq(1)

1, otherwise,

where the ct is the center of target bounding box on the feature map in the frame t . Eq (1) indicates that the samples are considered to be positive if they are within the radius R of the center of targetlocation on the feature map. R is set by users and fixed across all videos. Also, we weight the losses by the positive and negative samples to eliminate class imbalance issue.

4. DECOMPOSITION

The sampled examples are resized to 107×107×3 pixels. In this paper, the relu3 layer of the pretrained VGG-M [6] model is utilized as feature extractor because it captures more semantic part or category information than lower layers. For each example, the output 3×3×512 dimensional feature map is then fed into our classification network for final prediction. Our specially designed classification network consists of three 1 × 1 convolutional layers. The first two convolutional layers have 512 and 128 number of filters, respectively. The last one produces

2 outputs for target and background. For network learning, we again employs softmax logistic regression for simplicity. The target state in the current frame is given by finding the example with the maximum positive score as ext = arдmaxxi ∈Xt f

+(xi) [34], where f + is the output positive score from our classification network.

5. CONCLUSIONS

In this paper, we presented a theme that helpful for Deep Position -Specific Tracing (DPST), which decouples the single tracing task into a quantization task and a Decomposition task, and trains a special network for each task. The Quantization network exploits the Data from the current slice, and provides a Current position for state transition, which is helpful to improve the performance of tracing. Our Decomposition network has fewer trainable parameters due to its 1×1 convolutional layers and the global average pooling. This specially designed structure is less likely to over-fitting, and improves the tracing accuracy and robustness. The proposed DPST algorithm achieves competitive results on the benchmarks without using additional annotated Motioned Pictures, largely reducing the human effort for sticky tags.

REFERENCES

[1] Shai Avidan. 2007. Ensemble tracking. IEEE TPAMI 29, 2 (2007).

[2] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. 2009. Visual tracking with online multiple instance learning. In CVPR. IEEE, 983–990.

[3] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. 2016. Staple: Complementary learners for real-time tracking. In CVPR. IEEE, 1401–1409.

[4] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-Convolutional Siamese Networks for Object Tracking. arXiv preprint arXiv:1606.09549 (2016).

[5] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. 2010. Visual object tracking using adaptive correlation filters. In CVPR. IEEE, 2544–2550.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In BMVC. BMVA Press. arXiv:cs/1405.3531

[7] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In CVPR, Vol. 1. IEEE, 886–893.

[8] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. 2014. Accurate scale estimation for robust visual tracking. In BMVC. BMVA Press.

[9] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. 2015. Convolutional features for correlation filter based visual tracking. In ICCVW. IEEE, 58–66.

[10] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. 2015. Learning spatially regularized correlation filters for visual tracking. In ICCV. IEEE, 4310–4318.

[11] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. 2016. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In CVPR. IEEE, 1430–1438.

[12] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. 2016. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In ECCV. Springer, 472–488.

[13] Jia Deng,Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In CVPR. IEEE, 248–255.

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for eneric Visual Recognition.. In ICML. 647–655.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR. IEEE, 580–587.

[16] Helmut Grabner, Michael Grabner, and Horst Bischof. 2006. Real-time tracking via on-line boosting.. In BMVC, Vol. 1. BMVA Press, 6.

[17] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. 2016. Struck: Structured output tracking with kernels. IEEE TPAMI 38, 10 (2016), 2096–2109.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR. IEEE, 770–778.

[19] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2015. High-speed tracking with kernelized correlation filters. IEEE TPAMI 37, 3 (2015), 583–596.

[20] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. 2015. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network.. In ICML. 597–606.

[21] Zhibin Hong, Zhe Chen, ChaohuiWang, Xue Mei, Danil Prokhorov, and Dacheng Tao. 2015. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In CVPR. IEEE, 749–758.

[22] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder,Luka Čehovin, Tomas Vojir, Gustav Häger, Alan Lukežič, and Gustavo Fernandez et al. 2016. The Visual Object Tracking VOT2016 challenge results. Springer.(Oct 2016). http://www.springer.com/gp/book/9783319488806

[23] Matej Kristan, Jiri Matas, Aleš Leonardis, Michael Felsberg, Luka Čehovin, Gustavo Fernandez, Tomas Vojir, Gustav Häger, Georg Nebehay, and Roman Pflugfelder et al. 2015. The Visual Object Tracking VOT2015 challenge results. In Visual Object Tracking Workshop 2015 at ICCV2015. IEEE.

[24] Matej Kristan, Roman Pflugfelder, Aleš Leonardis, Jiri Matas, Luka Čehovin, Georg Nebehay, Tomas Vojir, Gustavo Fernandez, Alan Lukežič, and Aleksandar Dimitriev et al. 2014. The Visual Object Tracking VOT2014 challenge results. (2014). http://www.votchallenge.net/vot2014/program.html

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In NIPS. 1097–1105.

[26] A Li, M Lin, YWu, MH Yang, and S Yan. 2016. NUS-PRO: A New Visual Tracking Challenge. IEEE TPAMI 38, 2 (2016), 335–349.

[27] Hanxi Li, Yi Li, and Fatih Porikli. 2014. Robust online visual tracking with a single convolutional neural network. In ACCV. Springer, 194–209.

[28] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. arXiv preprint arXiv:1312.4400 (2013).

[29] Xiaobai Liu. 2016. V3I-STAL: Visual Vehicle-to-Vehicle Interaction via Simultaneous Tracking and Localization. In MM. ACM, New York, NY, USA, 1117–1126. https://doi.org/10.1145/2964284.2964285

[30] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. 2015. Hierarchical convolutional features for visual tracking. In ICCV. IEEE, 3074–3082.

[31] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. 2015. Longterm correlation tracking. In CVPR. IEEE, 5388–5396.

[32] Xue Mei and Haibin Ling. 2009. Robust visual tracking using l 1 minimization. In ICCV. IEEE, 1436–1443.

[33] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. 2016. Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint arXiv:1608.07242 (2016).

[34] Hyeonseob Nam and Bohyung Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In CVPR. IEEE, 4293–4302.

[35] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. 2016. Hedged deep tracking. In CVPR. IEEE, 4303– 4311.

[36] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. 2008. Incremental learning for robust visual tracking. IJCV 77, 1 (2008), 125–141.

[37] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR.

[38] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. 2014. Visual tracking: An experimental survey. IEEE TPAMI 36, 7 (2014), 1442–1468.

[39] Michael Stengel, Steve Grogorick, Martin Eisemann, Elmar Eisemann, and Marcus Magnor. 2015. An Affordable Solution for Binocular Eye Tracking and Calibration in Head-mounted Displays. In MM. ACM, 15–24.

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In CVPR. IEEE, 1–9.
[41] Ran Tao, Efstratios Gavves, and Arnold W M Smeulders. 2016. Siamese Instance Search for Tracking. In CVPR. IEEE.
[42] Andrea Vedaldi and Karel Lenc. 2015. Matconvnet: Convolutional neural networks for matlab. In MM. ACM, 689–692.
[43] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. 2013. Online object tracking with sparse prototypes. IEEE TIP 22, 1 (2013), 314–325.
[44] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. 2015. Visual tracking with fully convolutional networks. In ICCV. IEEE, 3119–3127.
[45] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. 2013. Online object tracking: A benchmark. In CVPR. IEEE, 2411–2418.
[46] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. 2015. Object tracking benchmark. IEEE TPAMI 37, 9 (2015), 1834–1848.
[47] Jianming Zhang, Shugao Ma, and Stan Sclaroff. 2014. MEEM: robust tracking via multiple experts using entropy minimization. In ECCV. Springer, 188–203.
[48] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. 2012. Real-time compressive tracking. In ECCV. Springer, 864–877.
[49] Gao Zhu, Fatih Porikli, and Hongdong Li. 2016. Beyond local search: Tracking objects everywhere with instance-specific proposals. In CVPR. IEEE, 943–951.