LEARNING ADAPTIVE SELECTION NETWORK FOR REAL-TIME VISUAL TRACKING

Jiangfeng Xiong, Xiangmin Xu, Bolun Cai, Xiaofen Xing, Kailing Guo*

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China
{xmxu, xfxing, guokl}@scut.edu.cn, {xiongjiangfeng, caibolun}@gmail.com

ABSTRACT

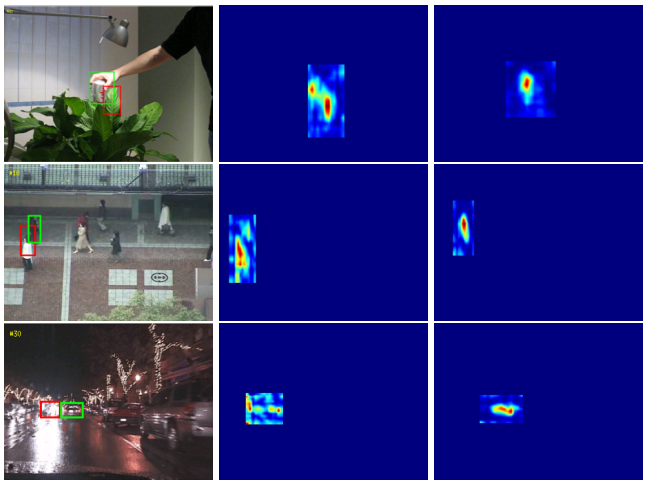
Offline-trained trackers based on convolutional neural networks (CNNs) have shown great potential in achieving balanced accuracy and real-time speed. However, offline-trained trackers are prone to drift to background clutters. In this paper, we present an adaptive selection network tracker (ASNT) to address the tracking drift problem. Inspired by feature selection technique used in other vision problems, we introduce a learnable selection unit for Siamese network based trackers. The selection unit enables the tracker to select relevant feature map automatically for the target. Channel dropout is applied in the selection unit to improve generalization performance for convolutional layers. To further improve the discrimination between background clutters and the target, an adaptive method is used to initialize the tracker for each video sequence. Experiments on OTB-2013 and VOT2014 datasets demonstrate that our ASNT tracker has a comparable performance against state-of-the-art methods, yet can run at a speed of over 100 fps.

Index Terms— online adaption, feature selection, real-time tracking

1. INTRODUCTION

Visual object tracking [1, 2, 3, 4] is a fundamental problem in computer vision with numerous applications, including motion analysis, video surveillance, human-computer interaction and robot perception. Given an object marked in the first frame, the goal of single-object tracker is to locate the selected object in the subsequent frames.

Convolutional neural networks (CNNs) have achieved great successes in a wide range of visual tasks [7, 8, 9]. Many works have been proposed to replace hand-crafted features with deep features in the traditional tracking framework, such as correlation filters [10, 11]. Ma et al. [10] propose to exploit different CNN layers and learn correlation filters on each layer to encode the target appearance. HDT [11] adaptively hedges features from different CNN layers in an online manner for visual tracking. The integration of deep features leads to a radical increase in the number of model parameters. Besides, correlation filters based trackers typically update the model in every frame. Although deep features vastly improve tracking performance of correlation filters based trackers, the factors described above result in slow tracking speed.



(a) Input (b) GOTURN [5] (c) ASNT (ours)

Fig. 1. Tracking results of GOTURN (red rectangles) and ASNT (green rectangles) for *coke*, *subway* and *cardark* sequences from [6]. We visualize the feature map of GOTURN and ASNT by channel average in (b) and (c), respectively. The feature map of GOTURN is noisy and it tends to drift to background clutters where large responses exist. In the feature map of ASNT, the large responses mainly exist within the target region, thus ANST achieves a more robust performance.

Recently, CNN-based methods [12, 13, 14, 15, 16, 5, 17, 18] have been proposed to learn the tracking models, which can be categorized into online-updated and offline-trained.

Online-updated trackers: Several online-updated trackers [12, 13] based on CNNs typically draw positive and negative training examples around the estimated target to incremen-

This work is supported by Natural Science Foundation of China (No. 61702192, No. 61751202 and U1636218) and Fundamental Research Funds for Central Universities (2017MS045).

* Corresponding author.

978-1-5386-1737-3/18/$31.00

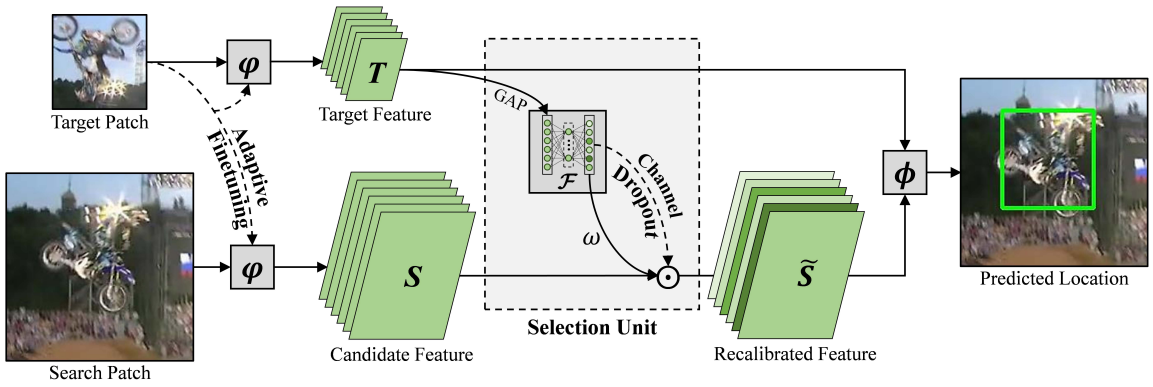


Fig. 2. Pipeline of the adaptive selection network tracker (ASNT). ASNT can be broken down into three constituent modules, namely feature representation φ, selection unit $\mathcal{F}$ and motion regression ϕ. (1) Feature representation φ is trained offline and fine-tuned adaptively in the initial frame. (2) Selection unit $\mathcal{F}$ performs dynamic feature recalibration with channel dropout to select more relevant features. (3) Motion regression ϕ compares the features from the target object and the selected features in the current frame to find where the target has moved.

tally learn a classifier. MDNet [12], the winner of the VOT 2015 challenge, learns a per-object classifier online. In [13], two CNNs are learned online for short-term and long-term appearance variation. FCNT [14] adopts the existing offline pre-trained network and introduces two convolutional layers to capture the appearance change with online updating. However, these methods often suffer from high computation burden and hardly run in real-time.

Offline-trained trackers: Some CNN-based trackers without online updating have been proposed, such as tracking by matching [15, 18] and tracking by location prediction [5, 17]. In [18], the SINT method generates a lot of particles and calculates their similarity scores using the Siamese network, which makes SINT run slowly. Bertinetto et al. [15] propose to learn a matching function with a fully convolutional Siamese network on a video object detection dataset. Since these trackers [15, 17, 5] are trained entirely offline, the networks are fast to evaluate at test time, allowing these methods to operate at faster than real-time speeds. However, the capabilities of offline trackers are limited because they cannot adapt to targeted information. GOTURN [5] as a real-time CNN-based tracker, trains the tracking model offline to regress the target location. Although GOTURN can run at high frame rates, it has lower tracking accuracy compared to the state-of-the-art trackers. The limits of GOTURN mainly come from two aspects. First, GOTURN benefits little from the end-to-end training since it borrows all the convolutional features pre-trained on ImageNet without finetuning. Second, GOTURN freezes all the weights of the tracking network and does not use any online adaption techniques. These two factors make GOTURN easily drifting to background clutters (shown in Fig. 1(b)). More recently, several works [16, 19] are proposed to address tracking drift for Siamese network based trackers. However, these methods could not estimate the scale of the target directly, the tracking speed is limited by the time-consuming evaluation on several scaled versions of the search image.

Based on GOTURN [5], we propose an adaptive selection network to address tracking drift and keep real-time performance (shown in Fig. 1(c)). Our main contributions are summarized as follows.

- First, a learnable selection unit is proposed to perform feature selection by highlighting relevant features and discounting less useful ones.
- Second, we apply channel dropout in the selection unit to reduce the correlation between learned features. Channel dropout forces the network to learn the features with better generalization.
- Third, an adaptive method is used to learn target-specific information, which greatly improves the tracking performance.

Extensive experiments on two popular benchmark datasets (OTB-2013 [6] and VOT2014 [20]) demonstrate that the proposed tracking algorithm can achieve state-of-the-art performance at high frame rates.

2. ADAPTIVE SELECTION NETWORK TRACKER

The adaptive selection network tracker (ASNT) proposed by us is illustrated in Fig. 2. We use the initial appearance of the object as target patch. The search patch is cropped centered at the previous position of the target. An identical transformation φ is applied on the target patch and the search patch to obtain the target feature $\boldsymbol{T}$ and the candidate feature $\boldsymbol{S}$, respectively. The feature map $\boldsymbol{S}$ of search patch is reweighted to generate the recalibrated feature map $\tilde{\boldsymbol{S}}$ by the selection unit.

In the end, the target location is regressed by the connection between $\boldsymbol{T}$ and $\tilde{\boldsymbol{S}}$. Detailed network settings of ASNT shown in Figure 2 are summarized in Table 1.

2.1. Selection Unit

The goal of feature selection is to highlight relevant features and suppress less useful ones. Inspired by SENet [**?**], we propose a learnable selection unit to explore the benefits of offline training. Different from SENet, we integrate target information into search stream to recalibrate the candidate feature as illustrated in Fig. 2. The selection unit can be broken down into two parts, including global average pooling and bottleneck structure.

To ensure efficiency, we use a much shallower network compared with the state-of-the-art architectures, where the reception field of the network is much smaller. Therefore, a global average pooling (GAP) is utilized to generate channel-wise statistics, which can fully exploit contextual information. Formally, the c-th element of statistic $\bar{\boldsymbol{T}} \in \mathbb{R}^C$ is given by

$$\bar{\boldsymbol{T}}_c = \frac{1}{W_t \times H_t} \sum_{i=1}^{W} \sum_{j=1}^{H} \boldsymbol{T}_c(i,j), \tag{1}$$

where $\boldsymbol{T} \in \mathbb{R}^{W_t \times H_t \times C}$ is the feature map of target patch.

A bottleneck structure is used to fully explore channel-wise dependencies. The bottleneck is made up of two fully connected layers with parameters $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, each layer is followed by batch normalization to stabilize and speedup training process. The derivation of the recalibrated feature map $\tilde{\boldsymbol{S}} \in \mathbb{R}^{W_s \times H_s \times C}$ can be formalized as

$$\tilde{\boldsymbol{S}} = f_c\left(\sigma\left(\boldsymbol{W}_2 \delta\left(\boldsymbol{W}_1 \bar{\boldsymbol{T}}\right)\right), \boldsymbol{S}\right), \tag{2}$$

where $\delta(\cdot)$ denotes ReLU activation function, $\sigma(\cdot)$ denotes Sigmoid activation function, and $f_c(\omega, \boldsymbol{S})$ refers to channel-wise multiplication between the search feature map $\boldsymbol{S} \in \mathbb{R}^{W_s \times H_s \times C}$ and the selection weight $\omega \in \mathbb{R}^C$.

2.2. Channel Dropout

CNNs can be regraded as an ensemble with each channel of the feature map to detect an individual pattern. The selection unit as a feature detector reinforces a certain type of visual patterns, where discriminative features are more important. In this paper, we propose channel dropout to increase the diversity of the visual patterns.

Dropout forces the network to learn interpretable features, which are less co-adapted and lead to better generalization. However, it has been demonstrated in [21] that standard dropout fails to reduce the correlation between learned features in convolutional layers and a spatial dropout is formulated to set all the values across the randomly selected channels of the feature map into zeros.

In our case, due to the channel-wise multiplication in the selection unit, channel dropout can be easily implemented by standard dropout in the selection weight ω. We can select the feature map randomly by performing channel dropout. Besides, by introducing channel dropout, our proposed ASNT is less likely to overfit than GOTURN [5]. In particular, the effectiveness of channel dropout is verified in the experimental section.

Table 1. The detailed architecture of ASNT. *K, C, S, P, D* refers to kernel size, channel numbers, stride, padding and dimension, respectively. All the convolutional layers and fully connected layers are followed by ReLU, except that FC2 adopts Sigmoid function and FC6 uses identity function.

Layer		Target Stream	Search Stream	Parameters
Feature Representation	Input	113×113×3	227×227×3	—
	Conv1	26×26×96	55×55×96	K11C96S4
	Pool1	13×13×96	27×27×96	K3S2
	Conv2	13×13×256	27×27×256	K5C256S1
	Pool2	6×6×256	13×13×256	K3S2
	Conv3	6×6×384	13×13×384	K3C384S1P1
	Conv4	6×6×384	13×13×384	K3C384S1P1
	Conv5	6×6×256	13×13×256	K3C256S1P1
	Pool5	—	6×6×256	K3S2
Selection Unit	GAP	↓ 256	↓	—
	FC1	↓ 16	↓	D16
	FC2	↓ 256	↓	D256
	Product	6×6×256		—
Motion Regression	Connect	6×6×512		—
	FC3	4096		D4096
	FC4	4096		D4096
	FC5	4096		D4096
	FC6	4		D4

2.3. Adaptive Learning

Ideally, the selection unit is able to remove the negative effects of background clutters. In practice, limited patterns are still inadequate to adapt arbitrary tracking sequence without learning target-specific information. Therefore, we apply an adaptive learning method to obtain more discriminative features for the tracking object.

As it has been analysed in GOTURN [5] that online training brings computational cost without substantial improvement. We argue that it is not appropriate to fine-tune motion regression ϕ, because it focuses on comparing two similar feature maps to predict the motion. Besides, we found that fine-tuning the fully connected layers tends to overfit, since there are massive parameters.

In this paper, we fix motion regression ϕ and update the other two modules (feature representation φ and selection unit $\mathcal{F}$) of ASNT in the initial frame. The parameters $\boldsymbol{\theta}$ of two top convolutional layers and the selection unit are updated by applying Adam solver:

$$\arg\min_{\boldsymbol{\theta}} \sum_{i}^{N} ||f(t_i, s_i, \boldsymbol{\theta}) - y_i||_1, \tag{3}$$

where f represents the whole network of ASNT, and N is the total number of training pairs. t_i and s_i refers to the target patch and the search patch sampled in the first frame respectively, and y_i denotes the corresponding ground-truth bounding box (upper-left and lower-right coordinates). To prevent overfitting, we adopt multi-scale sampling $\left[1, \sqrt{2}, 2\right]$ to augment training examples.

3. EXPERIMENTS

In this section, we compare the proposed ASNT with states-of-the-art trackers on two standard benchmark datasets, and analyse the effect of each component adopted in ASNT.

3.1. Implementation details

ASNT is trained using a combination of videos and images. We remove the overlaps with the test datasets, and use 296 remaining videos out of 314 videos in ALOV300+ [22] as training video sequences. We also leverage training set in ImageNet [7], which includes 478,807 objects with annotated bounding box. Different from [5], we sample two images from a training video that both contain the object at most 100 frames randomly when training, and advocate to use the target patch in the first frame when testing. Additionally, we augment these training examples using motion smooth [5] and color jittering (including randomly adjusting brightness, hue, contrast and saturation).

For offline training, the convolutional layers in ASNT are pre-trained on ImageNet [7]. We fine-tune the top two convolutional layers with a learning rate of 10^{-6} and fully-connected layers with 10^{-5}. The learning rate is divided by 10 every 500k iterations, and the channel dropout rate is set to 0.5. For adaptive learning, we fine-tune the network in the first frame, using Adam optimizer for 300 iterations with a learning rate of 10^{-4}. Our ASNT is implemented in C++ with *Caffe* framework, and runs at 137 fps on a single NVIDIA GeForce GTX 1080ti GPU.

3.2. Comparisons results

OTB-2013 dataset. We compare our ASNT with the trackers reported by OTB-2103 benchmark [6] and six more state-of-the-art trackers that can run in real-time (CFNet [16], SiamFC [15], Staple [23], GOTURN [5], KCF [3], DSST [24]). The

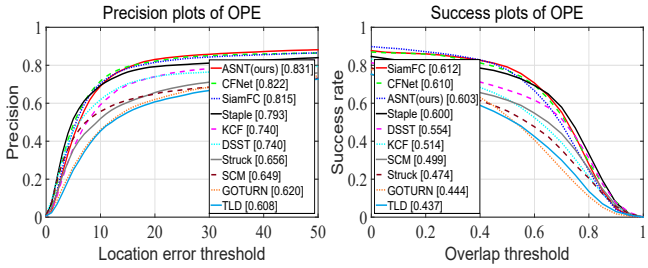


Fig. 3. Precision and success plots on OTB-2013. The scores in the legend indicate the mean precision when the location threshold is 20 pixels for the precision plots and area-under-curve (AUC) for the success plots.

tracking performance is measured by conducting a one-pass evaluation (OPE) based on precision score and success score. Fig. 3 shows the evaluation results. We only show the top 10 trackers for presentation clarity. Among all the trackers, the proposed ASNT performs favorably on the precision plots against state-of-the-art real-time trackers and achieves comparable results on the success plots. After tracking initialization in the initial frame, the proposed ASNT is able to track at 137 fps on a single GPU. However, for these tracking-by-verification methods [15, 16], they need to process several scale version of search image, which results in a much slower speed. Specially, the proposed ASNT is able to achieve a 21.1% and 15.9% improvement in terms of precession score and success score respectively compared to GOTURN [5]. The results clearly verify the superior tracking effectiveness and efficiency of our approach.

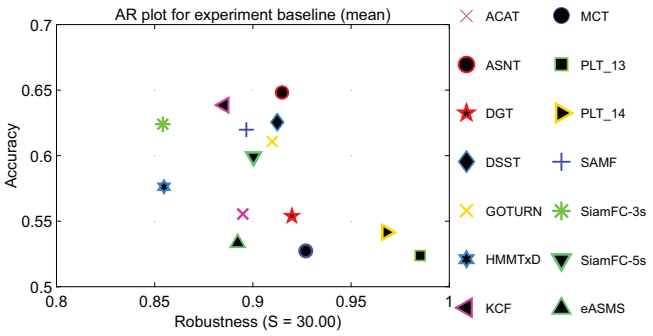


Fig. 4. VOT2014 Accuracy-Robustness plot. Best trackers are closer to the top right corner.

VOT-2014 dataset. We compare the proposed ASNT against the best 10 trackers that participated in the 2014 edition of the VOT challenge [20]. We also include two recent real-time Siamese network based trackers: GOTURN [5], SiamFC [15] (including two variants of SiamFC, SiamFC-5s and SiamFC-3s, which search over 5 scales and 3 scales, respectively). The trackers are evaluated using two standard tracking metrics: accuracy and robustness. Accuracy

Table 2. Average scores in percent of precision and success on different attributes: illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC) and low resolution (LR). The best results are shown in **red** and the second in blue. The last row shows the average speed of each tracker.

	TLD [1]	GOTURN [5]	SCM [25]	Struck [2]	DSST [24]	KCF [3]	Staple [23]	SiamFC [15]	CFNet [16]	ASNT (ours)
IV	53.7/39.9	52.8/39.9	59.4/47.3	55.8/42.8	73.0/56.1	72.8/49.3	74.1/**56.8**	71.7/54.2	72.1/53.1	**77.1**/55.6
SV	60.6/42.1	66.7/48.5	67.2/51.8	63.9/44.5	73.8/54.6	67.9/42.7	73.3/55.1	80.2/**60.3**	79.7/58.4	**82.5**/59.6
OCC	56.3/40.2	50.8/36.3	64.0/48.7	56.4/41.3	70.6/53.2	74.9/51.4	78.7/59.3	**79.7**/**59.4**	77.7/56.6	74.6/55.2
DEF	51.2/37.8	62.5/43.2	58.6/44.8	52.1/39.3	65.8/50.6	74.0/53.4	**81.2**/**61.8**	73.0/53.7	81.0/58.1	80.1/56.2
MB	51.8/40.4	45.8/34.4	33.9/29.8	55.1/43.3	54.4/45.5	65.0/49.7	68.8/54.1	72.6/**54.3**	67.6/53.5	**73.8**/53.1
FM	55.1/41.7	48.9/38.1	33.3/29.6	60.4/46.2	51.3/42.8	60.2/45.9	64.3/50.8	74.3/56.1	66.8/52.0	**76.4**/**57.3**
IPR	58.4/41.6	59.8/43.4	59.7/45.8	61.7/44.4	76.8/56.3	72.5/49.7	77.3/58.0	76.0/58.2	76.9/56.5	**85.2**/**60.1**
OPR	59.6/42.0	64.0/45.7	61.8/47.0	59.7/43.2	73.6/53.6	72.9/49.5	77.3/57.5	78.8/**58.8**	80.7/58.3	**81.2**/58.5
OV	57.6/45.7	45.0/37.4	42.9/36.1	53.9/45.9	51.1/46.2	65.0/55.0	67.9/54.7	**77.5**/**63.5**	44.3/42.3	71.0/55.5
BC	42.8/34.5	58.0/41.8	57.8/45.0	58.5/45.8	69.4/51.7	75.3/53.5	75.3/**57.6**	74.2/55.4	**77.0**/56.8	76.9/55.7
LR	34.9/30.9	36.8/27.1	30.5/27.9	54.5/37.2	49.7/40.8	38.1/31.2	55.0/43.8	**73.1**/**56.6**	55.3/43.4	60.8/46.1
Overall	60.8/43.7	62.0/44.4	64.9/49.9	65.6/47.4	74.0/55.4	74.0/51.4	79.3/60.0	81.5/**61.2**	82.2/61.0	**83.1**/60.3
Speed (fps)	28	165	0.5	20	24	172	80	58	67	137

is calculated as the average intersection-over-union (IOU), while robustness is expressed in terms of total number of failures. Within VOT2014 benchmark, trackers are automatically re-initialized five frames after failure. Fig. 4 shows the Accuracy-Robustness (AR) plot evaluated on VOT2014. The proposed ASNT outperforms all previous methods in VOT2014, GOTURN [5], and SiamFC [15]. Note that the proposed ASNT has a 4% improvement in terms of mean accuracy compared with GOTURN [5], which demonstrates the effectiveness of the proposed method.

3.3. Analysis

Contribution of each component. Since the original implementation of GOURN [5] uses some videos that overlapped with OTB-2013 [6] to train the network, we retrain GOTURN after removing the overlapped videos. Our implementation of GOTURN is slightly different in three ways. Firstly, we always crop the target patch in the first frame rather than the previous frame. Secondly, we crop the target patch without any background content. Thirdly, we adopt more augmentation methods as described in Section 3.1. We further use this implementation of GOTURN as our baseline (denoted as Base).

To verify the contribution of each component, we analyse the performance of four versions of ASNT on OTB-2013: Base, extension with selection unit (SU), channel dropout(CD), adaptive learning (AL) as shown in Fig. 5. Note that Base+SU+CD+AL is equivalent to the full version of ASNT. Especially, adaptive learning improves the discrimination between background clutters and the tracking object, which greatly enhances the performance. We found that ASNT without channel dropout (Base+SU+AL, which is not shown in Fig. 5 for presentation clarity) performs even worse than Base. We argue that it suffers from overfitting with more parameters. The experiment indicates that the proposed channel dropout is an important component in preventing overfitting in both offline training and adaptive learning.

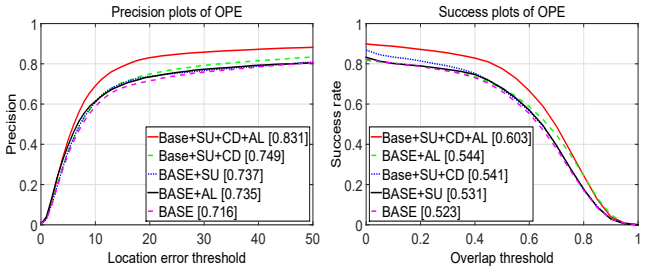


Fig. 5. Precision and success plots on OTB-2013 for the self-comparison of our algorithm.

Attribute-based analysis. In Table 2, we further analyse the tracking performance under different video attributes annotated in OTB-2013 [6]. The result indicates that the proposed ASNT is effective in handling illumination variation and fast motion. It is mainly because we adapt multi-scale sampling technique and color jittering augmentation as described in Section 3.1. Both GOTURN [5] and the proposed ASNT perform well in scale variation, this can be attributed to the motion smooth method used in offline training. For background clutters sequence, the proposed ASNT gains a significant improvement compared with GOTURN, this can be due to section unit and adaptive learning. The proposed ASNT utilizes fine-tuned CNN features, which are beneficial for accurate location especially under in-plane rotation and out-of-plane rotation situation.

4. CONCLUSION

In this work, we address the tracking drift problem for offline trained trackers by learning an adaptive selection network, while maintaining beyond real-time tracking speed. It is worth noting that the selection unit with channel dropout is very flexible and can benefit any other Siamese network based trackers. By interpreting ASNT as three constituent modules, namely feature representation, selection unit and motion regression, we propose an adaptive learning method, which proves to be effective in experiments. In our future work, we plan to integrate more background information into the selection unit to facilitate adaptive learning. We also consider exploring a lightweight architecture, which is of great importance for real-world tracking applications.

5. REFERENCES

[1] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[2] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr, "Struck: Structured output tracking with kernels," *TPAMI*, vol. 38, no. 10, pp. 2096–2109, 2016.

[3] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[4] Bolun Cai, Xiangmin Xu, Xiaofen Xing, Kui Jia, Jie Miao, and Dacheng Tao, "Bit: Biologically inspired tracker," *TIP*, vol. 25, no. 3, pp. 1327–1339, 2016.

[5] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," in *ECCV*. Springer, 2016, pp. 749–765.

[6] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[8] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[9] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao, "Dehazenet: An end-to-end system for single image haze removal," *TIP*, vol. 25, no. 11, pp. 5187–5198, 2016.

[10] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015, pp. 3074–3082.

[11] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang, "Hedged deep tracking," in *CVPR*, 2016, pp. 4303–4311.

[12] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302.

[13] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[14] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2015, pp. 3119–3127.

[15] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV*. Springer, 2016, pp. 850–865.

[16] Jack Valmadre, Luca Bertinetto, João F Henriques, Andrea Vedaldi, and Philip HS Torr, "End-to-end representation learning for correlation filter based tracking," *arXiv preprint arXiv:1704.06036*, 2017.

[17] Fei Zhao, Ming Tang, Yi Wu, and Jinqiao Wang, "Densetracker: A multi-task dense network for visual tracking," in *ICME*. IEEE, 2017, pp. 607–612.

[18] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016, pp. 1420–1429.

[19] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang, "Learning dynamic siamese network for visual object tracking," in *CVPR*, 2017, pp. 1763–1771.

[20] Matej Kristan, Roman P Pflugfelder, Ales Leonardis, Jiri Matas, Luka Cehovin, Georg Nebehay, Tomas Vojir, Gustavo Fernandez, Alan Lukezic, Aleksandar Dimitriev, et al., "The visual object tracking vot2014 challenge results," in *ECCVW*, 2015, pp. 191–217.

[21] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler, "Efficient object localization using convolutional networks," in *CVPR*, 2015, pp. 648–656.

[22] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah, "Visual tracking: An experimental survey," *TPAMI*, vol. 36, no. 7, pp. 1442–1468, 2014.

[23] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401–1409.

[24] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.

[25] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*. IEEE, 2012, pp. 1838–1845.