Online Recommendation-based Convolutional Features for Scale-Aware Visual Tracking

Ran Duan[†], Changhong Fu[†], Kostas Alexis and Erdal Kayacan[*]

Abstract— **In this paper, we develop an online learning-based visual tracking framework that can optimize the target model and estimate the scale variation for object tracking. We propose a recommender-based tracker, which is capable of selecting the representative convolutional neural network (CNN) layers and feature maps autonomously. In addition, the proposed recommender computes the weights of these layers and feature maps. A discriminative target percept of each recommended layer is reconstructed by the weighted sum of the recommended feature maps. Then the target model of the correlation filter is updated by the weighted sum of the target percepts. Thus, a sub-network is extracted from the pre-trained CNN backbone for the tracking process of a specific target. To deal with scale changes, we propose a spatiotemporal-based min-channel method to estimate the target size variation directly from CNN features. Experimental results on 50 benchmark datasets and video data from a rescue drone demonstrate that the proposed tracker is quite competitive with the state-of-the-art CNN-based trackers in terms of accuracy, scale adaptation, and robustness for UAV-related applications.**

Supplementary material

Open-source code with demo video:
`https://github.com/arclab-hku/ICRA2021tracking.git`

I. Introduction

Visual tracking, one of the fundamental problems in computer vision, has been widely used in numerous vision-based UAV applications [1], [2], [3], [4]. It is also one of the fundamental tasks in computer vision. Although being investigated for decades and much progress in terms of tracking accuracy and robustness has been made [5], [6], [7], [8], [9], [10], [11], [12], object tracking still remains a challenging problem due to many uncertainty factors, such as appearance variation, occlusion, background clutter. On the other hand, deep learning methods are good options to address this kind of uncertainty problem. Recently, convolutional neural networks (CNNs) have shown better performance in terms of accuracy and robustness for visual tracking task compared to state-of-the-art approaches [13], [14], [15], [7], [16], [17], [18], [9], [10], [6].

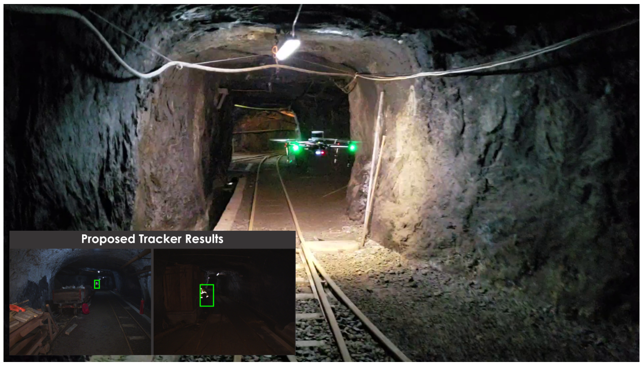

Fig. 1: Tracking task for the rescue drone.

Two-dimensional CNNs have exhibited outstanding performance in object recognition problems, for instance, YOLO [19], SSD [20], Faster RCNN [21], and Mask-RCNN [22]. In the backbone layers of those CNNs, an object can be represented by different levels of percepts, e.g., different complexity of the feature semantic combinations. Thus, a featureless object may have a better representation using low-level percepts while a complex object requires high-level percepts that contain highly discriminative information. This phenomenon is commonly referred to as the semantic gap. However, existing CNNs-based visual trackers use only one or several pre-selected layers [23], [13], [14], [15]. Furthermore, each feature map from hierarchical layers indicates the level of similarity of a specific type of feature throughout the whole image using image convolution. Therefore, taking all feature maps from a hierarchical layer is not a reasonable way to use CNN for object tracking since some of the features do not belong to the object. Another main challenge for CNN-based approaches is the scale variation. Since CNNs treat each frame as an independent image, continuity of object scale change is ignored. As the results, the CNN-based methods still have to use traditional methods for scale estimation of an untrained target. For instance, [13] solve the scale factor estimation in its updated version by extracting HoG feature in addition to CNN features.

In this work, the semantic gap issue mentioned above is addressed using a recommender for layer selection and updating appearance model using recommended feature maps. Hence the proposed method does not need the entire network in most small or featureless target tracking cases. Those efforts aim to reduce the work load for UAV onboard computer. The scale variation problem is solved by learning object scale directly from CNN features via a spatiotemporal-

[1] R. Duan is with Department of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, Hong Kong, China. ran-sn.duan@connect.polyu.hk

[2] C. Fu is with School of Mechanical Engineering, Tongji University, Shanghai, China. changhongfu@tongji.edu.cn

[3] K. Alexis is with University of Nevada, Reno, Nevada, USA & NTNU, Trondheim, Norway. konstantinos.alexis@ntnu.no

[4] E. Kayacan is with Artificial Intelligence in Robotics Laboratory (AiR Lab), the Department of Engineering, Aarhus University, 8000 Aarhus C, Denmark. erdal@eng.au.dk

[†] Ran Duan and Changhong Fu have contributed equally to this work.

[*] Corresponding author

based approach. In addition, the proposed scale learning framework also measures the certainty of tracking. Thus, the searching region grows together with the increasing of tracking uncertainty in the presence of target lost due to fast or abrupt motion of the drone platform. There are two fundamental novelties in our investigation:

- Automatic recommendation and weighting of the convolutional features that have appropriate feature semantic level and appearance representation of the target. Which allows the tracker to rebuild the appearance model of any untrained target and to simply the network.
- Scale estimation with searching region growing strategy that learns spatiotemporal variation of the target size directly from CNN feature maps and relocates the target to handle the blur or abrupt motion from drone view.

II. Related Work

In this section, three main tracking approaches, which are closely related to this work, are presented, i.e., tracking by correlation filters, tracking by CNNs, as well as their hybrid approaches.

Tracking by correlation filters: Correlation filters have gained considerable attention because they convert the problem into the Fourier domain. The trackers, which employ correlation filters, compute the regression between the circular-shifted input features, and a Gaussian function model refers to the target. A notable work [12], popularly known as kernelized correlation filters (KCF) demonstrated excellent tracking performance by combining multi-dimensional features and kernels and finding the best filter taps that maximize the correlation response of over-sampled target.

Tracking by CNNs: The research on CNN-based visual tracking has achieved remarkable performance. For instance, the DeepTrack [24] learns effective feature representations of the target object in a purely online manner. Hong et al. [23] take outputs from the first fully-connected layer to learn the target and background features. These approaches learn the positive and negative samples from a pre-trained CNNs. However, such models are designed to recognize numerous objects discard temporal information while the goal of visual tracking is to locate single or few objects' positions over time. Wang et al. [25] use a domain adaptation module for online adapt the pre-learned features according to the particular target object. This module has been integrated into other tracking methods and achieved significant improvement.

Tracking by KCF-CNN: Recently presented work, e.g. hierarchical convolutional features (HCF) [13] uses a KCF CNN-based hybrid approach. HCF uses 3 pre-selected layers, i.e., the max-pooling layers of $conv3$, $conv4$, and $conv5$, with fixed weights. The highest layer is able to discriminate the target while lower layers are used for precise localization. However, this method suffers from a complex background since most of the features that the CNN learned are background feature. In this paper, instead of using fixed layers, we propose a novel recommender to automatically select the best perceptive layers and the feature maps in each selected layer for the tracked object. To handle the scale variation, we present spatiotemporal-based min-channel feature maps. As a result, the target percept reconstructed from the recommended feature maps is robust to both appearance and scale changes of target objects.

III. Proposed Algorithm

In this section, we first give an overview of the proposed method, of which the framework is shown in Fig. 2. The target appearance is given in the first frame. Initially, the proposed recommender optimizes the VGGNet [26] network by finding the highest layer we actually need and build the optimized target model for the correlation filter training, which is discussed in Section III-A and Section III-B, respectively. In each new frame, we take a sample patch that is an extended region of the target region in the last frame and feed it to the optimized network - to extract the convolutional features. Then we use the proposed recommender to build the candidate model. The correlation filter works together with the proposed scale learning method to relocate the target and update the new model(Section III-C and Section III-D). Finally, we summarize the tracking framework using pseudocode.

A. Recommender

Denote $\bar{F}$ and $\bar{B}$ as the mean response value of foreground region F (given by bounding box) and background region B (the extended searching region) of a convolutional response $X(x_1, x_2, ..., x_N)$, respectively. The recommendation score of X is given by:

$$f(X, F, B) = DT \cdot GT, \tag{1}$$

where the DT and GT are distinctive term and gain term, respectively. They are defined as:

$$DT = \frac{1}{N}\sum_{i=1}^{N} x_i(e^{1-(\frac{2d_i}{(1+\beta)r})^2} - 1), \tag{2}$$

$$GT = (\bar{F} - \bar{B})^2, \tag{3}$$

where d_i is the pixel distance between i-th pixel to the target center and r is the diagonal length of region F. β is the tolerance parameter. x_i is the convolution response value of i-th pixel. As Fig. 3 shows, DT scores the feature quality, while GT measures the statistical difference between foreground and background. Fig. 4 shows the recommender output.

The high response of the convolution, i.e., $i \in \{i | x_i \geq \delta\}$, means the local appearance is highly correlated to the image filter (or kernel) that learned by CNNs. In this work, we are interested in the peak responses ($\delta = max(X)$) because our goal is single object tracking. For multiple objects tracking, the peak responses will be constrained by regions.

Remark 1: The typical value for tolerance parameter β is 0.25, which means we assume that the location shift or scale growing of the target is within 25% target size in general. It is introduced because the target may have a position shift or scale variance.

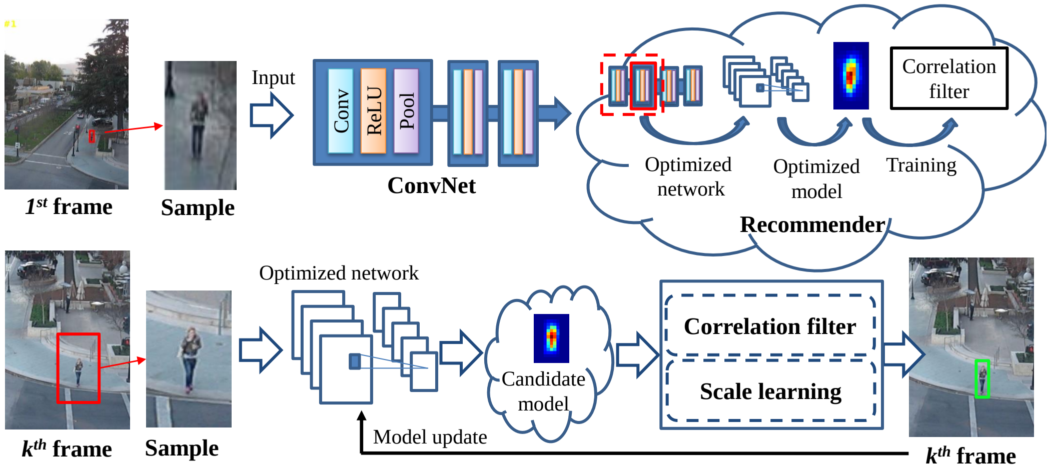

Fig. 2: Tracking framework overview

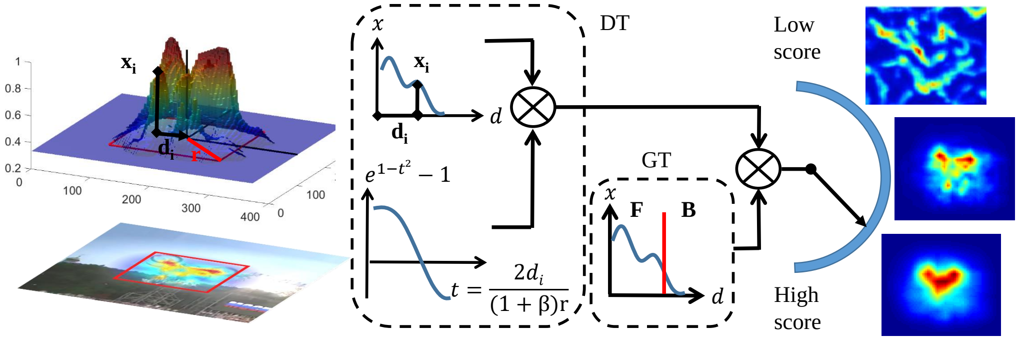

Fig. 3: Diagram of DT and GT. DT aims to select the discriminative features learned by CNN. As the new figure shows, for the convolution results, the peak values with a large distribution is designed to result in a low score. DT provides a convergent non-linear weight distribution refer to the $2d_i/(1+\beta)r$ ratio as well as labels foreground/background by the positive/negative score. GT, on the other hand, amplify the discriminative score by comparing the power of foreground and background.

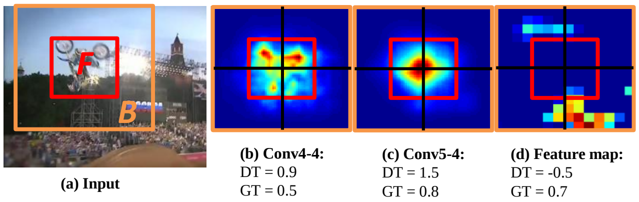

Fig. 4: Recommender output. The input in (a) shows the foreground (F) and background (B) regions. In (b), layer Conv4-4 has multiple dispersed peak convolutional responses, while in (c), the peak responses gather to the center. Therefore, compare to (b), the kernel learned in (c) is more likely to be the discriminative detector of the target. A sample of feature maps is shown in (d), of which the discriminative term DT is a negative value because it only represents background features. The gain term GT, on the other hand, measures the difference level between F and B.

B. *Target Appearance Modeling*

Figure 5 shows an example of target appearance modeling. We extract target percept C^j from the j-th convolutional layer by taking the average of its Gaussian weighted feature maps h_i^j:

$$C^j = \sum_i G \circ h_i^j, \tag{4}$$

where the G is a cosine window that weight the feature by Hadamard product (notated as $\circ$). This is used to avoid the discontinuity of image bounder. The h_i^j is the i-th feature map of j-th convolutional layer. Due to max-pooling, the image size of C^j varies. Therefore, each C^j is re-sampled with a fixed size. To be noted that a normalized C (values in the range [0,1]) is used for further computing.

In the first frame, we extracted C from all layers and compute their recommendation scores:

$$\mathbf{f}^c = \{f_j^c | f_j^c = f(C^j, F, B), \forall j\}, \tag{5}$$

by equation (1). The index set ϕ of recommended layers is given by:

$$\phi = \{j | f_j^c \in TopN(\mathbf{f^c})\}, \tag{6}$$

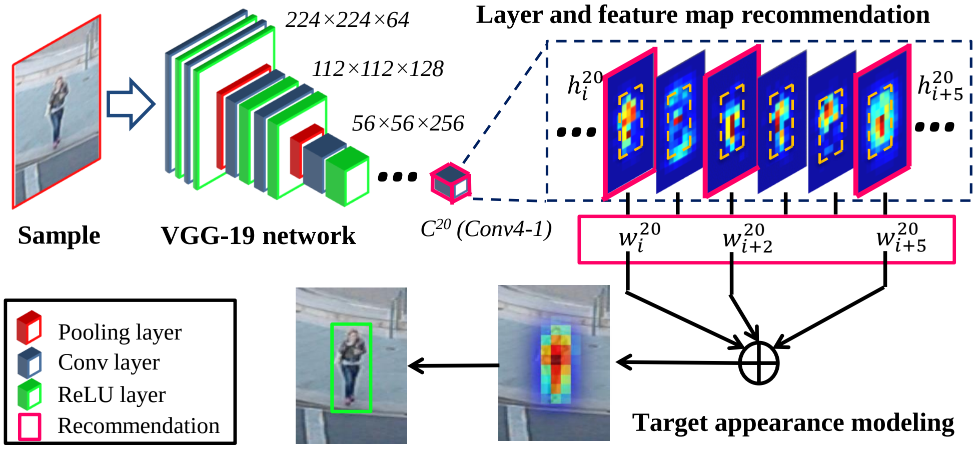

Fig. 5: Recommendation framework: The proposed recommender selects layer C^{20} weights and all feature maps in layer C^{20}. Only the feature maps with the positive weights are used to rebuild the target model. To be noted that the ReLU layers are taken into account, therefore there are 37 layers in VGG-19. Hence, we extract the sub-CNN with only 20 layers for this tracking task, which optimized the convolutional feature extraction process.

where function $TopN()$ returns the set of top N highest values of an input set.

Once ϕ is determined, we build the target model using recommended feature maps for each recommended layer. The recommendation scores of all feature maps in the j-th layer are computed as weights $\mathbf{w}^j = \{w_i^j | w_i^j = max(0, f(h_i^j, F, B), \forall i)\}$ (a recommended feature map h satisfy $f(h, F, B) > 0$). Then our reconstructed model $\mathbf{x}$ is define as:

$$\mathbf{x} = \{x^j | x^j = \sum_i w_i^j G \circ h_i^j, \forall j \in \phi\}. \tag{7}$$

Remark 2: In the CNN-based state-of-the-art trackers, such as our closest competitor HCF, the target percept is the weighted sum of the percepts obtained from pre-selected CNN layers by taking the sum of all feature maps in each layer. Since the input image patch could contain the background scene, the background features may also be updated to the target model. In our method, most of the background features are rejected by the proposed recommender and the features that represent the whole or critical parts of target dominate the result by giving higher weights. One example in the benchmark test is given in Fig. 6, which illustrates the importance of feature map recommendation by comparing the proposed method (without scale adaptation function) with HCF.

C. Correlation Filters

Correlation filters are trained by the linear regression method. Denote the vectorized samples of target x^n and a vectorized 2D Gaussian window $y \sim \mathcal{N}(\mu, \sigma^2)$, where μ is the center of target sample and σ is the kernel width. To be noted that y can also be the data labels because the weights in window y indicate the distances to the target center. Then the linear regression problem is formed by:

$$\arg\min_{\omega} \|\mathbf{x}\omega - y\|^2 + \lambda\|\omega\|^2, \tag{8}$$

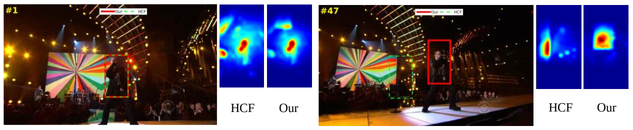

Fig. 6: With (our: red bounding box) or without (HCF: green bounding box) feature selection by the proposed recommender. In order to do a fair comparison, we disabled the scale adaptation of our method. In this example, the background scene contains more features than the tracked object. As the CNNs simply detect any learned features, without the recommendation of the feature maps, the background features may dominate the target appearance model, resulting in the drifting of tracking.

where ω is the coefficients to be trained, $\mathbf{x} = \{x^1, x^2, ...\}$, I is an identity matrix and λ is the regularization coefficient. The closed-form solution of 8 is given by:

$$\omega = (\mathbf{x}^T\mathbf{x} + \lambda I)^{-1}\mathbf{x}^T y. \tag{9}$$

To speed up the process, we train ω in Fourier domain. In the first frame, the target model $\mathbf{X}$ is built by taking Fourier transforms of reconstructed feature $\mathbf{X} = \mathcal{F}(\mathbf{x})$ and $\mathbf{Y} = \mathcal{F}(\mathbf{y})$. The initial trained correlation filter $\mathbf{W}$ is define as:

$$\mathbf{W} = \{W^j | W^j = \frac{Y \circ \bar{X}^j}{X^j \circ \bar{X}^j + \lambda}, \forall j \in \phi\}, \tag{10}$$

where the $\bar{X}$ denote the complex conjugate of X.

From the second frame, our correlation filter estimates target location by computing the weighted correlation response R:

$$R = \mathcal{F}^{-1}(\sum_j f_j' W^j \circ X^j), \forall j \in \phi, \tag{11}$$

where f_j' is the normalized score over recommendation score $\mathbf{f}^c$ from equation (5) with range [0,1]. Therefore, the

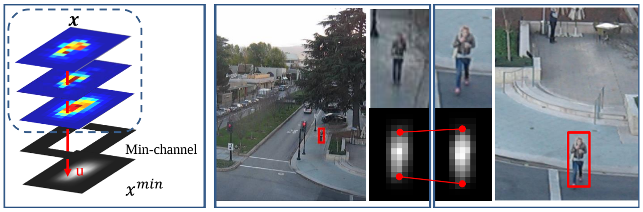

Fig. 7: Spatiotemporal-based min-channel scale: the first block on the left describes the main idea of min-channel. The highest responses of all recommended feature maps are projecting into the min-channel map, in which we only take the pessimistic of target region into consideration. Since the highest response of feature maps are the location of target features, the variation of power distribution of the min-channel maps are used for target scale change estimation.

local location u^* with maximum correlation response is the estimated new location of the target:

$$u^* = \arg\max_u R(u). \tag{12}$$

After the target is relocated, we extract the CNN feature again in order to learn the target appearance online. Let t be the index of frame sequence and α be the learning rate, the updating process of numerator $\mathbf{A}$ and denominator $\mathbf{B}$ of filter W can be written as:

$$\mathbf{A}_t = (1-\alpha)\mathbf{A}_{t-1} + \alpha Y \circ \bar{\mathbf{X}}_t, \tag{13}$$

$$\mathbf{B}_t = (1-\alpha)\mathbf{B}_{t-1} + \alpha \mathbf{X}_t \circ \bar{\mathbf{X}}_t, \tag{14}$$

$$\mathbf{W}_t = \frac{\mathbf{A}_t}{\mathbf{B}_t + \lambda}. \tag{15}$$

D. Min-channel Scale Learning

In general, we assume the target size changes continuously in visual tracking task. Therefore, for scale estimation, temporal information should be able to increase the accuracy of scale estimation. In this work, we propose a spatiotemporal-based min-channel scale learning scheme. The min-channel is a binary mask that crops the minimum target region (bounding box region) from the searching region.

The min-channel approach aims to reject the noise as much as possible. To obtain the min-channel map of the target, we project the max values at each location u throughout all recommended feature maps, i.e., find the maximum value on the third dimension of feature map set $\mathbf{x}$. Then corp it by the min-channel mask MC.

$$x^{min}(u) = MC(u) \cdot max(\mathbf{x}(u)), \forall u. \tag{16}$$

Then the new scale s at frame t refer to the first frame is updated as:

$$s_t = \gamma \frac{\bar{s}\sum_u x_t^{min}(u)}{\sum_u x_1^{min}(u)} + (1-\gamma)\frac{\sigma_t^w}{\sigma_1^w}, \tag{17}$$

where σ^w is a weighted standard deviation of x^{min}. The average scale of a short term memory $\bar{s}$ is computed after applying median filter to $\{s_{t-1}, s_{t-2}, ..., s_{t-M-1}\}$, where M is the memory size. A typical value for γ is 0.9.

Remark 3: Because the feature maps extracted from CNN are re-sampled with a fixed size, $\bar{s}$ is used to recover the true statistical characteristics of x_t^{min}. For the same reason, term $\frac{\sigma_t^w}{\sigma_1^w}$ does not affect the result when the previous scale estimation is correct. Wrong updating of target percepts tend to expand the distribution of significant response of x^{min} since they are mainly background features. Therefore, the purpose of introducing this term to extend the searching region when the uncertainty (the distribution of high convolutional response) of estimation increased.

E. Tracking Framework

We denote input t-th frame I_t, target region F_t and its extended searching region B_t. Function $CNNs()$ extracts the feature maps of input image. The tracking framework of proposed tracker is shown in algorithm 1.

Algorithm 1: Tracking framework of proposed tracker

```
Data: $\{I_t | t = 1, 2, ...\}, F_1$
Result: $\{F_t | t = 2, 3, ...\}$
initialization: $B_1$, $t \leftarrow 1$, $s_1 \leftarrow 1$ ;
$\mathbf{h} \leftarrow CNNs(I_t(B_t))$ // Extract CNN features
$\phi \leftarrow Eq.\ (6)(\mathbf{h})$ // Index set of recommended layers
$\mathbf{x_t} \leftarrow Eq.\ (7)(\mathbf{h}, \phi, F_t, B_t)$ // Target percept
$\mathbf{W}_t \leftarrow Eq.\ (10)(\mathbf{x_t})$ // Correlation Filter
$x_t^{min} \leftarrow Eq.\ (16)(\mathbf{x_t})$ // Min-channel map
while $t <$ frame length do
    $t \leftarrow t + 1$
    $\mathbf{h} \leftarrow CNNs(I_t(B_{t-1}))$
    $\mathbf{x} \leftarrow Eq.\ (7)(\mathbf{h}, \phi, F_{t-1}, B_{t-1})$
    $[F_t, B_t] \leftarrow Eq.\ (11)\&\ (12)(\mathbf{W}_{t-1}, \mathbf{x})$
    $x_t^{min} \leftarrow Eq.\ (16)(\mathbf{x})$
    $s_t \leftarrow Eq.\ (17)(x_t^{min}, x_1^{min}, \{s_{t-1}, ..., s_{t-M-1}\})$
    $[F_t, B_t] \leftarrow ScaleUpdate(F_t, B_t, s_t)$
    $\mathbf{h} \leftarrow CNNs(I_t(B_t))$
    $\mathbf{x}_t \leftarrow Eq.\ (7)(\mathbf{h}, \phi, F_t, B_t)$
    $\mathbf{W}_t \leftarrow Eq.\ (15)(\mathbf{W}_{t-1}, \mathbf{x}_t)$
end
```

IV. Experiments

In this section, we demonstrate the performance of the proposed tracker by Visual Tracker Benchmark v1.0 test. The benchmark protocol is proposed by Y. Wu, *et al* [27]. We tested our tracker on 50 datasets and evaluated with the following three evaluation methods: **Center location error (CLE)** measures the tracking accuracy by computing the distance between the centers of the ground truth and the estimated bounding box. **Success rate (SR)** measures the overlap between the ground truth and bounding box, which is related to scale adaptation. CLE and SR are also denoted as mean average precision (mAP) and intersection over union (IoU) in object detection research.

Remark 4: We run our tracker as well as other 10 state-of-the-art trackers on each dataset from the first frame to

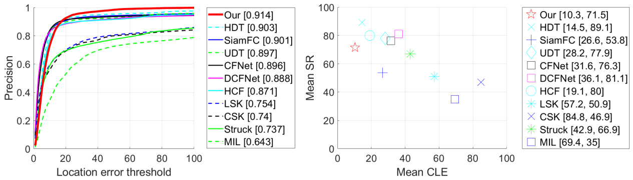

Fig. 8: Location precision plot (CLE: pixel-based distance) of OPE using AUC and the merged plot of mean CLE and SR. This figure shows the overall performance of trackers in terms of tracking accuracy and robustness. The proposed tracker achieves the highest precision and the best average CLE (10.3).

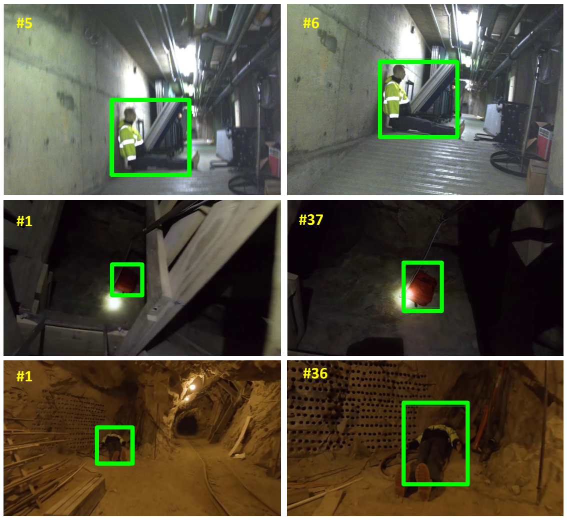

Fig. 9: During the flight, image blur and abrupt motion happens occasionally. The illuminance and target scale also variate over time.

the end, referred to a one-pass evaluation (OPE). Then the final tracking results for each dataset is obtained by taking the average of 10 times running. The final mean results are calculated from all frames over the 50 tested datasets.

Our test results are compared with other 10 high performance state-of-the-art trackers: HCF [13], SiamFC [14], CFNet [15], Struck [7], DCFNet [16], HDT [17], UDT [18], CSK [9], LSK [10], and MIL [6].

A. *Overall Performance*

We choose top 2 best layers and use the typical values for correlation filter parameters: $\alpha = 0.01$, $\lambda = 10^{-4}$. The overall performance is shown in Fig. 8 using the area under curve (AUC) and merged plot of mean CLE and mean SR, respectively.

The benchmark evaluation results illustrate that the proposed tracker performs competitively good tracking results in both tracking accuracy and scale adaptation compare to other 10 top-ranked state-of-the-art trackers. To be noticed that accurate tracking may also generate large CLE when the target is big. Furthermore, the tracker may generate random CLE when it gets lost. The second best CLE is given by tracker HDT. In our experiments, we simply use the top 2 recommended layers for target percept reconstruction. Although high layer carries very poor location information, our optimized target searching region and percept reconstruction significantly improved the tracking precision and tiny discontinuity of position shifting is observed in some datasets. In addition, we performed the proposed tracker on video data recorded by a rescue drone. The drone was flying inside the building or cave. The tracking targets include survivors and their belonging such as a bag. The video data contains blurred, fast motion, and discontinuous frames due to UAV platform. In Fig.9, we illustrate the tracking performance under blur and abrupt motion, illuminance change and large scale variation. Demo video is available on Github page.

B. *Results Analysis*

The results indicate that the proposed method outperforms other 10 state-of-the-art trackers by average tracking accuracy while SR is the second echelon ranked by overall evaluation. In Fig. 8 we can find that the proposed method is not top ranked when the threshold is less than 20. This indicates a drawback of the proposed tracker: the poor feature location information when only high-level CNN layers are used. The low layer features have more precise location information but less discriminate to background features. Unlike HCF which uses fixed hierarchical features to guarantee the location information, the proposed algorithm sometimes may only use high layers of CNN, which is a tradeoff between tracking accuracy and robustness. In conclusion, the proposed method achieves competitive overall performance against other 10 top-ranked state-of-the-art trackers. The major time consumption comes from the CNN feature extraction because the complexity of the correlation filter and the proposed recommender are $\Theta(n^2)$ and $\Theta(n)$, respectively. While the CNN goes to $\Theta(\sum_{l=1}^{D} M_l^2 K_l^2 C_{l-1} C_l)$, where the M is the length of the feature map, K is the length of the Kernel, C is the number of channels in each layer, l is the current layer and D is the number of the layers. The proposed recommender computes the highest D that is needed for a target in the first frame, which means we could have a smaller D for some tasks, resulting in a speedup of CNN feature extraction. Using VGG19 Matlab model, it achieves around 15FPS on the laptop with 1050Ti GPU. In future work, we will replace VGG19 net with other backbone, such as VGG16 (SSD), darknet (YOLO), or mobileNet, for UAV onboard tracking.

V. Conclusion

In this paper, we proposed a novel CNN-based tracker that simplifies the network and learns a quality appearance model with scale estimation using a recommender for the untrained target. Experimental results on 50 challenging benchmark datasets and drone recorded data demonstrated that the proposed method achieves competitive performance against 10 top-ranked state-of-the-art trackers in terms of tracking accuracy, scale adaptation, and tracking robustness.

References

[1] C. Fu, A. Carrio, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, “Robust real-time vision-based aircraft tracking from Unmanned Aerial Vehicles,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5441–5446.

[2] M. Mueller, G. Sharma, N. Smith, and B. Ghanem, “Persistent Aerial Tracking system for UAVs,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1562–1569.

[3] C. Fu, R. Duan, D. Kircali, and E. Kayacan, “Onboard Robust Visual Tracking for UAVs Using a Reliable Global-Local Object Model,” *Sensors*, vol. 16, no. 9, 2016.

[4] H. Chen and P. Lu, “Computationally efficient obstacle avoidance trajectory planner for uavs based on heuristic angular search method,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020, pp. 5693–5699.

[5] R. Duan, C. Fu, and E. Kayacan, “Recoverable recommended keypoint-aware visual tracking using coupled-layer appearance modelling,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4085–4091.

[6] B. Babenko, M.-H. Yang, and S. Belongie, “Robust Object Tracking with Online Multiple Instance Learning,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[7] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 263–270.

[8] C. Fu, R. Duan, and E. Kayacan, “Visual tracking with online structural similarity-based weighted multiple instance learning,” *Information Sciences*, vol. 481, pp. 292 – 310, 2019.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the Circulant Structure of Tracking-by-detection with Kernels,” in *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ser. ECCV’12. Springer-Verlag, 2012, pp. 702–715.

[10] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, “Robust tracking using local sparse appearance model and K-selection,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1313–1320.

[11] R. Duan, C. Fu, E. Kayacan, and D. P. Paudel, “Recommended keypoint-aware tracker: Adaptive real-time visual tracking using consensus feature prior ranking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 449–453.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[13] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, “Hierarchical convolutional features for visual tracking,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3074–3082.

[14] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” *arXiv preprint arXiv:1606.09549*, 2016.

[15] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for correlation filter based tracking,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] J. X. M. Z. W. H. Qiang Wang, Jin Gao, “Dcfnet: Discriminant correlation filters network for visual tracking,” *arXiv preprint arXiv:1704.04057*, 2017.

[17] Y. Qi, S. Zhang, L. Qin, Q. Huang, H. Yao, J. Lim, and M. Yang, “Hedging deep features for visual tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1116–1130, May 2019.

[18] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, “Unsupervised deep tracking,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 21–37.

[21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.

[23] S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” in *Proceedings of the 32nd International Conference on Machine Learning, 2015, Lille, France, 6-11 July 2015*, 2015.

[24] H. Li, Y. Li, and F. Porikli, “DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.

[25] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, “Video Tracking Using Learned Hierarchical Features,” *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, April 2015.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[27] Y. Wu, J. Lim, and M. Yang, “Object Tracking Benchmark,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 9, pp. 1834–1848, 2015.