electronics

MDPI

Article

Fully Convolutional Single-Crop Siamese Networks for Real-Time Visual Object Tracking

Dong-Hyun Lee

Department of IT Convergence Engineering, Kumoh National Institute of Technology, 61 Daehak-ro, Gumi 39177, Gyeongbuk, Korea; donglee@kumoh.ac.kr

Received: 22 August 2019; Accepted: 21 September 2019; Published: 24 September 2019

Abstract: The visual object tracking problem seeks to track an arbitrary object in a video, and many deep convolutional neural network-based algorithms have achieved significant performance improvements in recent years. However, most of them do not guarantee real-time operation due to the large computation overhead for deep feature extraction. This paper presents a single-crop visual object tracking algorithm based on a fully convolutional Siamese network (SiamFC). The proposed algorithm significantly reduces the computation burden by extracting multiple scale feature maps from a single image crop. Experimental results show that the proposed algorithm demonstrates superior speed performance in comparison with that of SiamFC.

Keywords: visual object tracking; deep learning; convolutional neural networks; Siamese networks

1. Introduction

The objective of visual tracking is to track an arbitrary object in a video, where the ground-truth bounding box is given in the first frame [1–3]. It is one of the essential computer vision tasks for many application areas such as video surveillance, autonomous navigation, and self driving [4–6]. The visual object tracking problem is challenging because the appearance of the target object is given in the first frame only. Moreover, the motion of the object and camera in the video incur appearance change, motion blur, occlusion, scale change, and poor illumination, which make the problem much harder.

There are two major approaches for visual object tracking: correlation filter-based approaches and deep convolutional neural network-based approaches. The tracking algorithms based on the correlation filter train a regressor from a set of training samples by updating the weights of filters, and the operation is performed in the Fourier domain using the fast Fourier transform [7–9]. Although the correlation filter-based methods are computationally efficient, they suffer from boundary effects, which cause an inaccurate representation of image content. This reduces the discriminative ability of the trained filter and limits the target search region. Recently, deep features have been utilized for correlation filter-based approaches to track more robust and semantic features of the targets [10–12]. However, since the correlation filters must be updated during tracking and the training of the convolutional neural networks requires heavy computation, they suffer from speed degeneration. The deep convolutional neural network-based approaches take advantage of end-to-end training by applying convolutional neural networks (CNNs) for deep feature extraction and object localization [13–15]. These can be divided into online and offline approaches depending on the training methodology. In the case of the online training-based trackers, the target object is trained during test time [16,17]. Although the domain specific information of the target is continuously updated during tracking, their tracking speed is quite slow since CNNs require heavy computation in training. Thus, most of them are not applicable for real-time tracking operation. In the case of the offline training-based trackers, the networks are trained offline from a large number of available video datasets [18]. However, since the model is not updated during tracking, they are not robust in a large deformation and occlusion of the target object.

In recent years, fully convolutional Siamese network (SiamFC) and its variations have gained increased attention due to their high performance in tracking speed and accuracy [19–21]. However, SiamFC requires multi-scale tests for scale estimation such that image patches with different scales per each image frame must be tested independently. This causes an increase of computation time proportional to the number of image patches. In order to reduce computation overhead while sacrificing little in precision, this paper proposes a single-crop SiamFC-based tracking framework. The proposed approach uses only a single-cropped patch per each image frame to generate the feature maps of multiple scales with less computation. This significantly improves tracking speed of SiamFC with a negligible precision loss.

2. Related Work

A Siamese network consists of two CNN branches for extracting features from two images, and one neural network to compare the features [22]. Due to its computational efficiency in estimation and its excellent performance in accuracy and robustness, it has been widely used in various applications such as face recognition, image matching, and one-shot recognition [23–25]. The Siamese networks have also been widely used in visual object tracking [19,26,27]. GOTURN uses AlexNet [28] as two CNN-based feature extraction branches. The feature maps from the previous and current image frames are fed to the fully connected layers to estimate the bounding box of the target [26]. In Re^3, a recurrent neural network is applied to the Siamese network to improve tracking performance when the target is occluded by other objects [27]. However, both approaches use fully connected layers to combine the CNN branches, and thus require large number of training parameters as well as heavy computation. SiamFC, on the other hand, only uses a fully convolutional neural networks and introduces a novel cross-correlation layer to connect the two branches [19]. SiamFC computes the similarity between multiple search images and the target exemplar image, which is cropped from the padded ground-truth bounding box from the first image frame. The output is a scalar-valued score map and the position of the tracking object is determined by selecting the location with the maximum score.

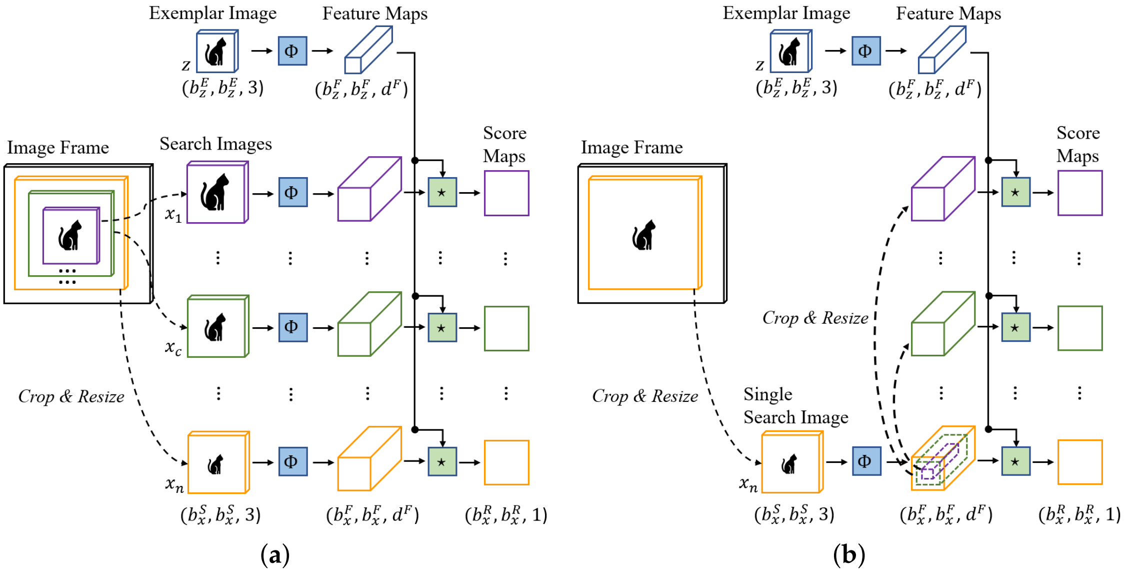
The SiamFC has drawn considerable attention in visual object tracking community as the CFNet, which incorporates the SiamFC with the correlation filter, won the Visual Object Tracking (VOT) 2017 real-time challenge [14,29]. There are many visual object tracking architectures that use the SiamFC as the baseline network. MBSiam combines SiamFC with a bounding box regression network, which uses SSD-MobileNet [30,31]. By adding the bounding box regression network, MBSiam can estimate tight bounding box of the target object. SiamRPN consists of two sub-networks, one for feature extraction and the other for region proposal, and two branches, a classification branch and regression branch [32]. For online tracking, the correlation layers are formulated as convolutional layers. DaSiamRPN is an extended version of SiamRPN for long-term tracking with effective sampling strategy to control the imbalance sample distribution [33]. SA-Siam uses two Siamese networks, a semantic branch for high-level feature extraction and an appearance branch for low-level feature extraction [34]. The two branches are trained separately to keep their heterogeneity and the estimation results from two branches are combined after the similarity score from each network is obtained. SiamVGG is based on the structure of SiamFC and extract features from a target patch and the search region by using a modified VGG-16 network [35]. SINT combined the optical flow with SiamFC to achieve better performance by focusing more on moving objects [36].

Most of the SiamFC-based approaches combine additional algorithms, such as a box regression network and optical flow, to improve accuracy and robustness of SiamFC. However, they still require multiple scales of search images for robust scale estimation of the target object. As a result, although they outperform the original SiamFC with respect to accuracy and robustness in object tracking, they require higher computation time for their additional algorithms as well as multi-scale processing. Such computational overhead makes them less practical for real-time applications that use embedded systems. The proposed single-crop SiamFC framework in this paper differs from aforementioned algorithms in that it only uses a single-cropped image patch in each frame and creates multiple feature maps with different scales to minimize computation time while maintaining the performance quality of SiamFC.

3. Single-Crop SiamFC Framework

3.1. Framework Overview

The feature map extraction process of baseline SiamFC and the proposed framework are shown in Figure 1a,b, respectively. In the case of SiamFC, multiple search images are generated by cropping different scales of image patches as shown in Figure 1a. The search images are separately fed into the convolutional embedding function, which is denoted as Φ, to generate the feature maps with different scales. Since the feature extraction process in Φ takes most of the computation time, the inference process time is increased in proportional to the number of search images.



(**a**) (**b**)

Figure 1. The feature map extraction process of fully convolutional Siamese network (SiamFC) (**a**) and the proposed framework (**b**).

In the case of the proposed approach, on the other hand, only uses a single search image as shown in Figure 1b. Inspired by the region proposal network in the Faster R-CNN [37], which extracts proposals directly from the feature map instead of the image, the proposed framework crops feature maps with different scales and resizes them to generate different-scale feature maps. Although multiple feature map cropping strategy could reduce some amount of tracking precision, it significantly speeds up the tracking process, which is essential for embedded vision applications. The rest of the process, such as cross correlation with the feature map from the exemplar image and bounding box estimation from the score maps, are the same as the baseline SiamFC.

3.2. Multi-Scale Feature Extraction

For the bounding box of the target with the size (w, h) in the image frame, the size (width, height) of the bounding box for the exemplar image, (b_z^I, b_z^I) is defined as

$$b_z^I = \sqrt{(w+2p)(h+2p)} \tag{1}$$

where $p = (w+h)/4$ is the amount of context margin around the target. The cropped image patch is resized to the fixed size as (b_z^E, b_z^E), which is defined as the exemplar image z. As shown in Figure 1a, the SiamFC searches the target in multiple scales by cropping different sized image patches. The *i*th size of the bounding box for the image patch with n number of different scales, $(b^I_{x_i}, b^I_{x_i})$, is defined as

$$b^I_{x_i} = b^I_x \alpha^{m(i)} \tag{2}$$

with

$$b^I_x = \frac{b^S_x}{b^F_z} b^I_z, \qquad m(i) = i - \frac{n+1}{2}, \qquad i \in \{1, \dots, n\} \tag{3}$$

where b^I_x is the size of the image crop with scale 1, b^S_x is the fixed size of the search image, and α is the scaling step. All the n cropped image patches are resized to (b^S_x, b^S_x) and used as the search images, represented as $x_1, \dots, x_n$. As shown in Figure 1a, each of the search images pass through a convolutional embedding function Φ such that the total n number of feature maps, $\Phi(x_1), \dots, \Phi(x_n)$, with a fixed size of (b^F_x, b^F_x), are generated. This causes the linear increase of feature map computation with respect to the number of scales. By contrast, the proposed approach only uses a single-crop from the image frame, which corresponds to the search image of the largest scale, x_n, and computes its feature map, which is $\Phi(x_n)$. The rest of the feature maps with different scales are cropped directly from $\Phi(x_n)$ and resized to (b^F_x, b^F_x) as shown in Figure 1b. The *i*th bounding box to crop the *i*th of feature map from $\Phi(x_n)$, $(b^F_{x_i}, b^F_{x_i})$, is defined as

$$b^F_{x_i} = b^F_x \alpha^{i-n} \tag{4}$$

where $i \in \{1, \dots, n\}$. After cropping, all the cropped feature maps are resized to (b^F_x, b^F_x). Each of the n number of feature maps from the search images and the feature map of the exemplar image are cross-correlated ($\star$ operation in Figure 1). This generates the n number of scalar-valued score maps, where each has the size of (b^R_x, b^R_x). The scale of the target is determined by selecting the score map which has the highest score value among the score maps. Then, the crop size of the exemplar and search images for the next frame, b'^I_z and b'^I_x are updated as

$$b'^I_z = (1-\gamma) b^I_z + \gamma b^I_z \alpha^{i^\star} b'^I_x = (1-\gamma) b^I_x + \gamma b^I_x \alpha^{i^\star} \tag{5}$$

where $i^\star$ represents the index of the score map that contains the maximum score value, and γ $(0 \leq \gamma \leq 1)$ is the update rate. Similarly, the width and height of the target in the next image frame, w' and h', are updated as

$$w' = (1-\gamma) w + \gamma w \alpha^{i^\star} h' = (1-\gamma) h + \gamma h \alpha^{i^\star}. \tag{6}$$

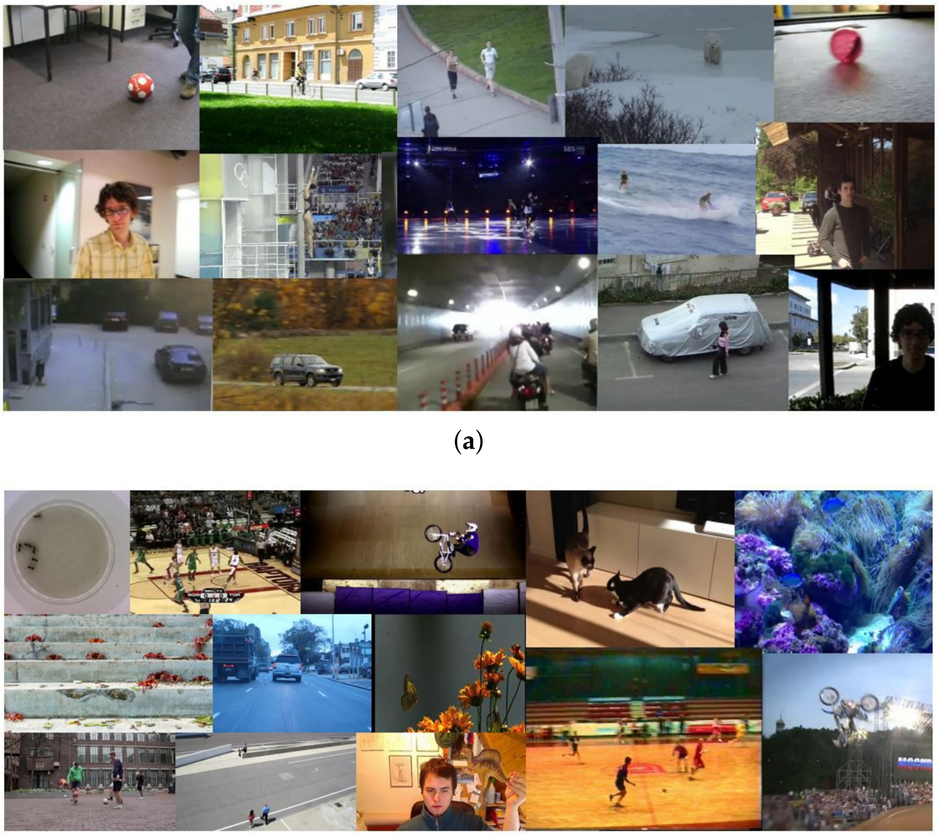
From the 2D location of the maximum score in the score map, the location of the target in the next image frame, x' and y', are updated as

$$x' = x + \beta^\star (b^R_x/2 - r^\star_x) y' = y + \beta^\star (b^R_y/2 - r^\star_y) \tag{7}$$

where $r^\star_x$ and $r^\star_y$ are the x and y locations of the maximum score in the score map, respectively, and $\beta^\star = b_{(}x_{i^\star})^I / b^R_x$ is the scaler that converts from the score map to the cropped image with the scale of $\alpha^{i^\star}$.

4. Experiments

The proposed algorithm was implemented with Python 3.6.5 and TensorFlow 1.9.0 by modifying the source code by the authors of SiamFC [19]. Both SiamFC and the proposed algorithm used the same pretrained network from [19]. In the experiment, the Visual Object Tracking (VOT) 2014 challenge dataset (25 videos) and VOT 2018 short-term (ST) challenge dataset (60 videos) were used [3,38]. The VOT 2015, 2016, and 2017 dataset heavily overlap with the VOT 2018 dataset, therefore, only the VOT 2014 and 2018 datasets were used in the experiment. As shown in Figure 2, the two datasets consist of sequences of various objects, such as ants, balls, and sports players. Each video was divided from one to six subsequences in the evaluation to correctly measure the tracking performance. For the evaluation, four quantitative metrics, precision plot, area under curve (AUC), intersection over union (IoU), and speed, were used. The precision plot represents the success frame rate (Number of success frames/Number of total frames) with respect to the pixel location error threshold. The success frame was increased by one at each frame if the pixel distance between the tracked target center and the ground-truth position was less than the location error threshold. The AUC represents the area under the curve of the precision plot and the IoU is the area of overlap over the area of union between the tracked and the ground-truth bounding boxes. The algorithm speed represents the number of frames that the algorithm can process per second (fps). The experiments were conducted with both CPU-only and with-GPU cases. For CPU and GPU, Intel i7-6800K at 3.4GHz and NVIDIA Geforce GTX 1080Ti were used, respectively.

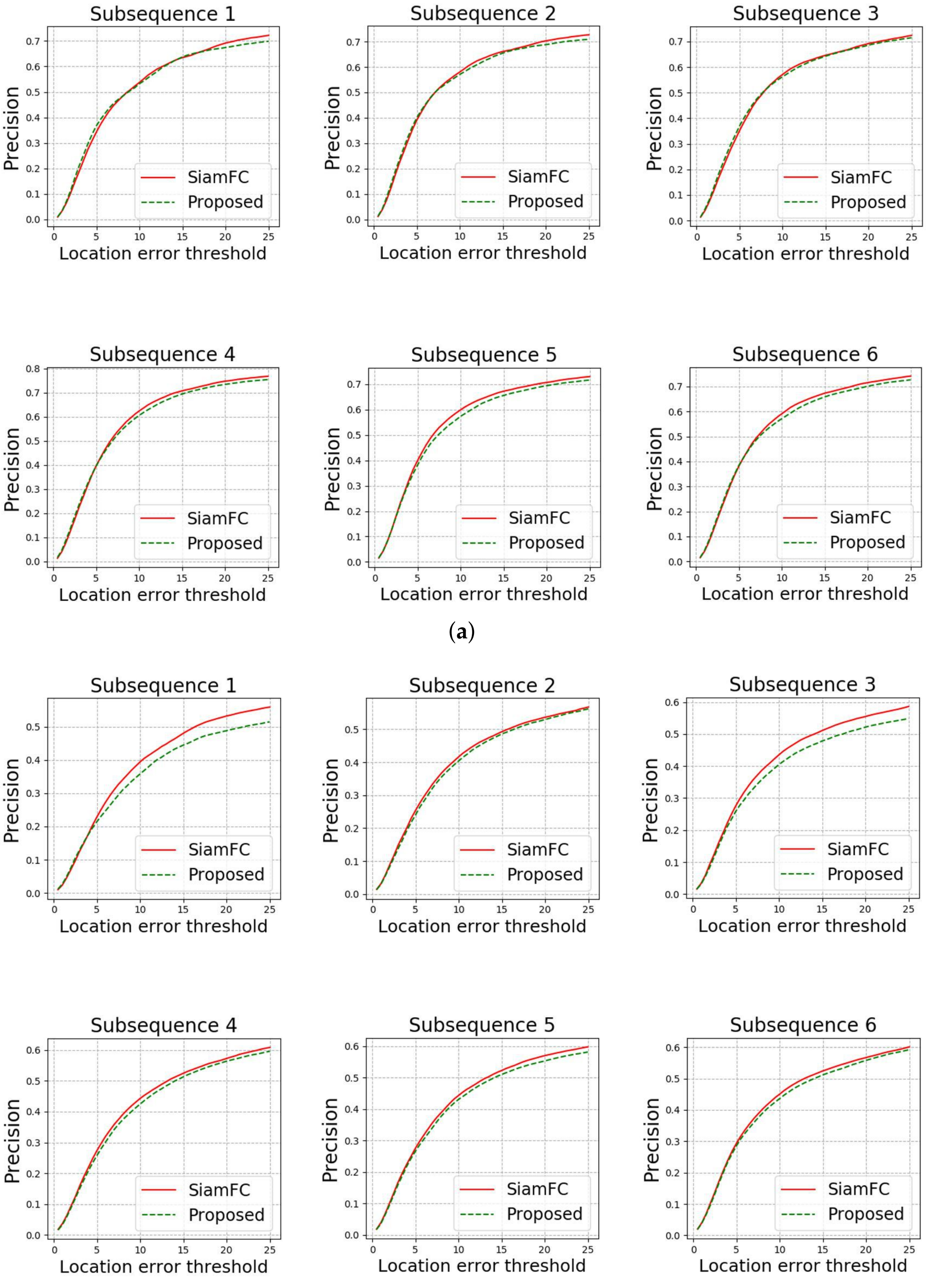


(**a**)

(**b**)

Figure 2. Samples from the Visual Object Tracking (VOT) 2014 (**a**) and 2018 (**b**) challenge datasets.

The precision plots of VOT2014 and VOT2018 over the six subsequences are shown in Figure 3a,b, respectively. The x axis is the location error threshold, which is from 0.5 to 25 pixels with the pixel gap of 0.5, and the y axis is the average success frame rate of the algorithms over all the videos in the dataset. As shown in the graphs, the tracking precision between SiamFC and the proposed algorithm is almost the same. Tables 1 and 2 show the precision (with the local error threshold of 20 pixels), AUC, and IoU of VOT2014 and VOT2018 over the six subsequences, respectively. The results from Figure 3, Tables 1 and 2 indicate that extracting different scales of feature maps from the largest scale feature map insignificantly degenerates the tracking performance with respect to precision, AUC, and IoU.



(**a**)

(**b**)

Figure 3. The precision plot of (**a**) VOT2014 and (**b**) VOT2018.

Table 1. The precision, area under curve (AUC), and intersection over union (IoU) of VOT2014.

Subsequence	Measure	SiamFC	Proposed
1	Precision	71.85	69.81
	AUC	25.88	25.79
	IoU (%)	53.79	50.08
2	Precision	73.22	69.86
	AUC	27.14	26.43
	IoU (%)	54.34	51.39
3	Precision	70.47	68.86
	AUC	25.98	25.83
	IoU (%)	54.30	51.26
4	Precision	74.83	71.84
	AUC	27.94	27.07
	IoU (%)	56.34	52.82
5	Precision	70.56	67.24
	AUC	26.63	25.39
	IoU (%)	54.64	50.79
6	Precision	71.49	67.96
	AUC	26.41	25.33
	IoU (%)	54.33	51.39

Table 2. The precision, AUC, and IoU of VOT2018.

Subsequence	Measure	SiamFC	Proposed
1	Precision	51.48	51.11
	AUC	18.43	17.91
	IoU (%)	34.17	32.10
2	Precision	53.61	54.41
	AUC	19.45	19.28
	IoU (%)	36.93	35.02
3	Precision	52.36	49.83
	AUC	19.01	17.80
	IoU (%)	37.05	33.56
4	Precision	53.95	53.79
	AUC	19.69	19.10
	IoU (%)	37.82	35.45
5	Precision	53.58	51.53
	AUC	19.44	18.54
	IoU (%)	37.84	34.82
6	Precision	54.16	52.35
	AUC	19.68	18.78
	IoU (%)	38.80	35.55

The average algorithm speed is shown in Figure 4. In the case of using only the CPU, the proposed algorithm is about 2.2 times faster than SiamFC. The results demonstrate that the proposed algorithm significantly improves the tracking speed with little tracking precision loss. In the case of using the GPU, the proposed algorithm is about 1.3 times faster than SiamFC. The GPU can be used for parallel computing of multiple feature maps, so the speed improvement is less than the CPU case. However, for the embedded systems where GPUs are not available for computing deep features from the search images, the proposed algorithm is more applicable than SiamFC. Figure 5 shows six snapshots of the videos in the VOT2018 ST dataset and the bounding boxes of the proposed algorithm. As shown in the figure, the proposed algorithm adjusts the scale changes of the objects during tracking.

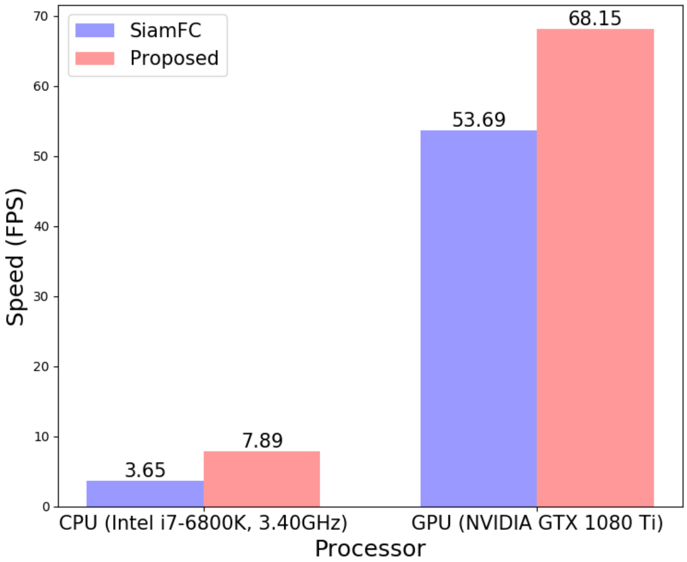


Figure 4. The tracking speed from CPU and GPU.

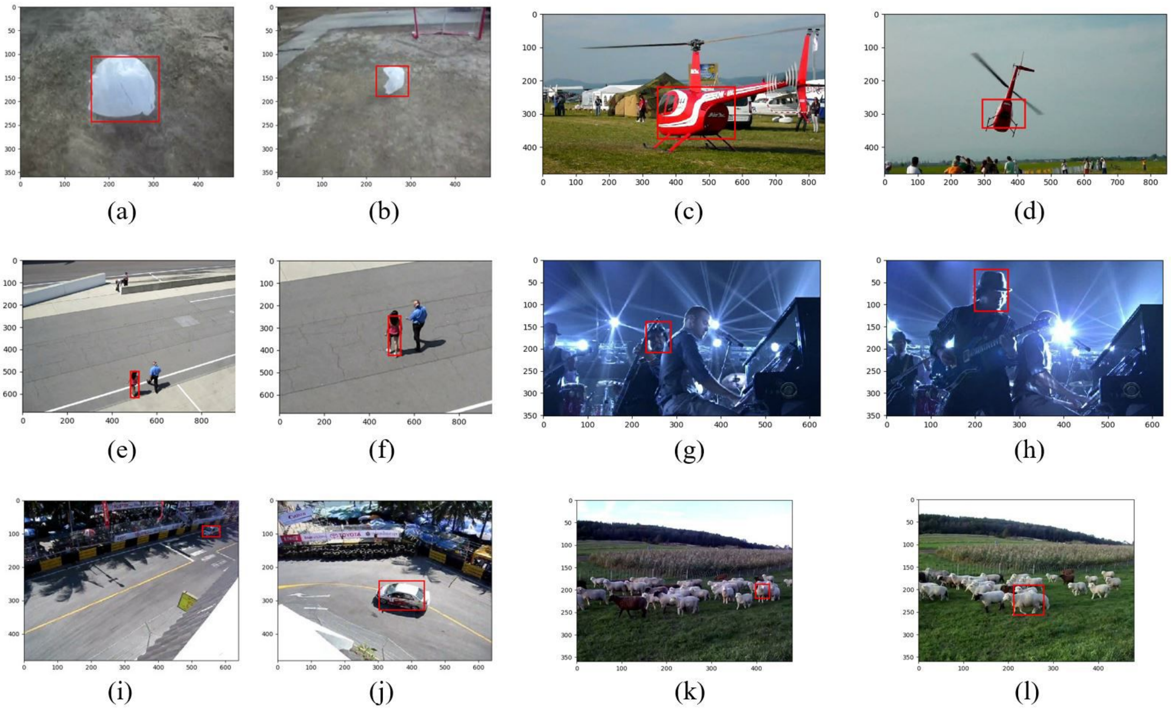


Figure 5. Screenshots of the tracking result from the proposed algorithm from VOT2018 dataset: bag (**a**,**b**), helicopter (**c**,**d**), crossing (**e**,**f**), shaking (**g**,**h**), racing (**i**,**j**), and sheep (**k**,**l**).

5. Conclusions

In this study, a single-crop Siamese network was proposed to increase the speed of SiamFC in visual object tracking. Instead of computing deep feature maps of multiple image crops for scale estimation, the proposed algorithm only uses a single image crop and extracts multiple scales of feature maps from a single deep feature map. The experimental results with VOT2014 and VOT2018 ST demonstrate that the proposed algorithm significantly improves the performance speed with little precision loss. The proposed work can be applied to various SiamFC-based algorithms to improve their speed. For future work, the proposed algorithm will be implemented on the embedded boards for visual object tracking.

Funding: This research was supported by Kumoh National Institute of Technology (2017-104-159).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
2. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
3. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; Pflugfelder, R. The visual object tracking vot2015 challenge results. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 1–23.
4. Li, S.; Yeung, D.Y. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
5. Erol, B.A.; Majumdar, A.; Lwowski, J.; Benavidez, P.; Rad, P.; Jamshidi, M. Improved deep neural network object tracking system for applications in home robotics. In *Computational Intelligence for Pattern Recognition*; Springer: Berlin, Germany, 2018; pp. 369–395.
6. Lee, K.H.; Hwang, J.N. On-road pedestrian tracking across multiple driving recorders. *IEEE Trans. Multimed.* **2015**, *17*, 1429–1438. [CrossRef]
7. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
8. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
9. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
10. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 58–66.
11. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
12. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488.
13. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
14. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
15. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Stct: Sequentially training convolutional networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1373–1381.
16. Wang, N.; Shi, J.; Yeung, D.Y.; Jia, J. Understanding and diagnosing visual tracking systems. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3101–3109.
17. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1409–1422. [CrossRef] [PubMed]

18. Fan, J.; Xu, W.; Wu, Y.; Gong, Y. Human tracking using convolutional neural networks. *IEEE Trans. Neural Netw.* **2010**, *21*, 1610–1623. [PubMed]
19. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
20. Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; Hu, W. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv* **2017**, arXiv:1704.04057.
21. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
22. Pflugfelder, R. An in-depth analysis of visual tracking with siamese neural networks. *arXiv* **2017**, arXiv:1707.00569.
23. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
24. Lin, T.Y.; Cui, Y.; Belongie, S.; Hays, J. Learning deep representations for ground-to-aerial geolocalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5007–5015.
25. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 10–11 July 2015; Volume 2.
26. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.
27. Gordon, D.; Farhadi, A.; Fox, D. Re^3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. *IEEE Robot. Autom. Lett.* **2018**, *3*, 788–795. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1097–1105.
29. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Hager, G.; Lukezic, A.; Eldesokey, A.; et al. The visual object tracking vot2017 challenge results. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1949–1972.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016, pp. 21–37.
32. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
33. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
34. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold siamese network for real-time object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4834–4843.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 420–1429.

37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2015; pp. 91–99.

38. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Zajc, L.Č.; Vojir, T.; Bhat, G.; Lukežič, A.; Eldesokey, A.; et al. The sixth visual object tracking vot2018 challenge results. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–53.