Robust and Real-Time Visual Tracking with Triplet Convolutional Neural Network

Jung Uk Kim
Image and Video Systems Lab
KAIST
Daejeon, Korea
jukim0701@kaist.ac.kr

Hak Gu Kim
Image and Video Systems Lab
KAIST
Daejeon, Korea
hgkim0331@ kaist.ac.kr

Yong Man Ro
Image and Video Systems Lab
KAIST
Daejeon, Korea
ymro@ kaist.ac.kr

ABSTRACT

In this paper, we propose a new visual object tracking which realizes robustness against object occlusion and deformation. In the proposed visual tracking, triplet convolutional neural network (triplet-CNN) structure is devised. The three inputs for the triplet-CNN come from current query frame, tracked object in a previous frame, and reference object. Object location in the query frame is predicted by fusing latent features from the three inputs. Moreover, predicted object is compared with reference object by using a Siamese CNN, so that object occlusion and deformation are detected and search range of tracking object is found adaptively. Comprehensive experimental results on a large-scale benchmark database showed that the proposed method outperformed state-of-the-art tracking methods in terms of precision and robustness with real-time tracking (about 25 fps).

KEYWORDS

Visual tracking; deep learning; occlusion; deformation

1 INTRODUCTION

Visual tracking has been attracting a significant interest in pattern recognition and computer vision due to the wide range of applications including video surveillance [1], activity analysis [2], video communication [3], and human-computer interaction [4]. Despite a lot of studies in recent years [5,6], it still remains a challenging task in practice due to many different and varying circumstances such as occlusion, deformation, background clutter, illumination variation, etc. [5]. In particular, severe object occlusion and deformation are one of the most critical factors that cause the drift of the tracker. In addition, real-time requirement is the one of the important issue in practice in various applications.

To address the problems for robust visual tracking, many visual tracking methods were proposed. Adam et al. proposed a fragments-based robust tracking method using integral histogram [5]. In [5], by combining fragments-based representation and voting maps, they could deal with partial occlusions or pose variation. Authors of [6] proposed multiple people tracking method using part-based model and dynamic occlusion handling. In [6], the part-based person-specific support vector machine (SVM) classifiers were learned to detect and track the human bodies in changing appearance and background. In [7], long-term visual tracker was proposed to consistently track the object with large appearance variations. The long-term tracker consisted of scale estimation and translation to detect target objects. However, the existing methods were limited to deal with partial occlusions only because they used ensemble of hand-craft features extracted from each part [5,6]. In addition, it is difficult to design the hand-craft feature for generic objects and their variations [7].

Recently, deep learning has attracted significant attentions due to their successful performance by learning hierarchical visual features. There have been increasing amount of works on deep learning-based visual tracking [8–11]. Results in there works indicate that a deep neural network, especially convolutional neural network (CNN), are effective to learn discriminative feature of input data, so that it is useful to enhance the performance of the visual object tracking. In [8], a visual tracking method using fully convolutional network was proposed. In [9], multi-domain learning based on CNN was proposed. It helped to discriminate target object from background. On the other hand, these methods require a large amount of computational cost to process the object candidate with deep convolutional neural network. It has limitations in real-time object tracking. In [10], fast generic object tracking method using deep regression networks was proposed. In this method, one search region in the current frame was compared with the predicted target in the previous frame to predict the object location fast. However, this method is insufficient to deal with drift when object is wrongly tracked. In [11], hierarchical convolutional features were extracted for visual tracking to encode various appearance feature of the object. In this method, by adopting multi-level correlation filters on convolution layer, they encode appearance of the object to find the object location. Although many methods have been reported in visual object tracking using deep learning, severe object occlusion and deformation remain a challenge in practice.

The aim of this paper is to propose new deep learning based visual tracking framework that provides both robustness and real-

ThematicWorkshops'17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-5416-5/17/10...$15.00
DOI: h.p://dx.doi.org/10.1145/3126686.3126739

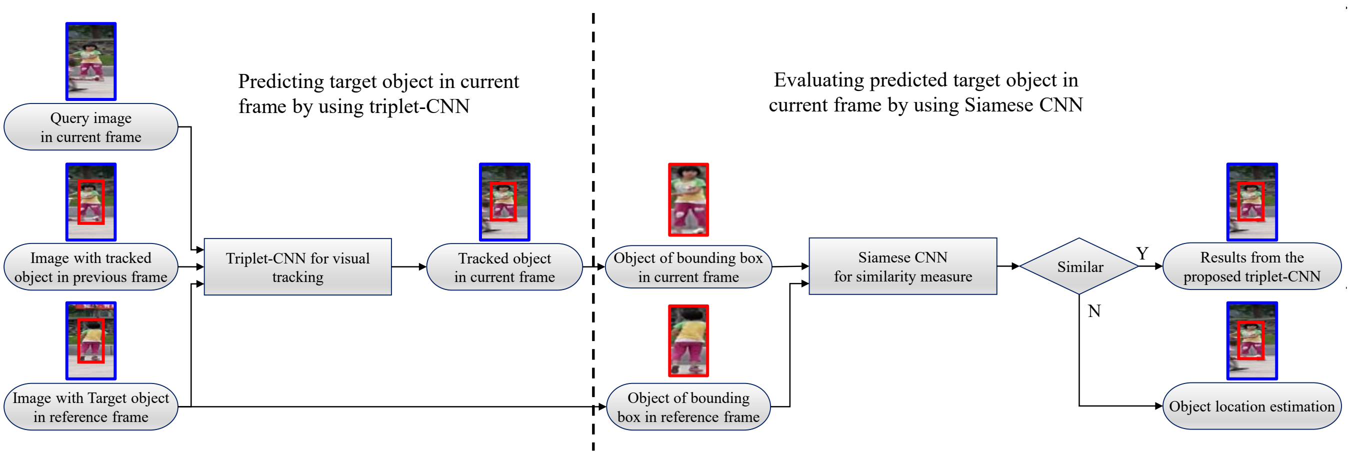

Predicting target object in current frame by using triplet-CNN
Evaluating predicted target object in current frame by using Siamese CNN
Query image in current frame
Image with tracked object in previous frame
Image with Target object in reference frame
Triplet-CNN for visual tracking
Tracked object in current frame
Object of bounding box in current frame
Object of bounding box in reference frame
Siamese CNN for similarity measure
Similar
Y
N
Results from the proposed triplet-CNN
Object location estimation

Figure 1: Overall procedure of the proposed deep visual tracking framework robust to occlusion and deformation. Proposed triplet-CNN directly regresses to the coordinates of the target object using appearances information of the previous and reference frames. Then, the tracked object from the triplet-CNN and the target object in reference frame are fed into Siamese CNN.

time against object occlusion and deformation. The main contributions of our paper are twofold:

1) This paper proposes new deep visual tracking framework that is robust to object occlusion and deformation. To track the target object in various challenging circumstances, triplet-CNN for visual tracking is devised. The triplet-CNN learns different appearance information of target object. Moreover, it directly predicts the location of object from the query image of the current frame rather than a large amount of computational cost to process many object candidates. As a result, the proposed method is able to track target object with about 25 fps.
2) The predicted location of the object could be inaccurate in severe occlusion or large deformation cases. To prevent drift or wrong object tracking in challenging conditions, in this paper, the predicted object is further evaluated by a Siamese CNN to check object occlusion and deformation. When detecting occlusion or severe deformation, the location of object is estimated from object motion information.

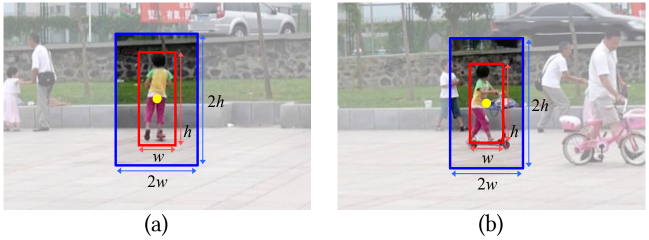
The rest of the paper is organized as follows. Section 2 presents the proposed robust visual tracking using triplet-CNN and Siamese CNN. In section 3, experimental results are presented to verify the performance of the proposed visual tracking. The conclusions are drawn in section 4.

2 PROPOSED METHOD

Fig. 1 shows the overall procedure of the proposed visual tracking. As shown in the figure, the proposed visual tracking consists of the two parts. In the first part, the location of the target object in current frame is predicted by using triplet-CNN. In the triplet-CNN, appearance changes of object are taken into account in learning the triplet-CNN with three inputs (query image in current frame, image with tracked object in previous frame, and image with target object in reference frame).

In the second part, the predicted object location (i.e., predicted bounding box in the current frame) is evaluated and adjusted as necessary. The reference object (i.e., ground-truth bounding box in the reference frame) as well as the predicted object is fed to Siamese CNN. The Siamese CNN measures similarity between the predicted object and the reference object so that object occlusion and deformation are detected.

If the similarity value is high, the location of the target object can be the result of the triplet-CNN. Otherwise, the location of target object is estimated from previous tracked objects. Detailed descriptions of each part are given in the following sections.



(a) (b)

Figure 2: Examples of cropped region used as inputs of triplet-CNN for various appearances. (a) Bounding box and cropped region in reference frame. (b) Bounding box and cropped region in previous frame. Red and blue boxes indicate the object-bounding box and cropped region, respectively. Yellow dot indicates the center position of the object-bounding box. Cropped region is two times of object-bounding box.

2.1 Triplet-CNN for visual tracking

To learn latent appearance features of the target object, we devise a triplet-CNN architecture. To consider appearance changes of the target object, three inputs, which come from target object in reference frame, the tracked object in previous frame, and current query frame, are fed to the triple-CNN. To represent the target object well, surrounding regions of the object in each frame are included in inputs of the proposed triple-CNN. Let (b_{cx}, b_{cy}) denote a center position of the object-bounding box. w and h denote the width and height of the bounding box (i.e., the size of the target object), respectively.

Fig. 2 shows examples of cropped region, which is used as input images for various appearance of target object. Fig. 2(a) and 2(b) show the regions in a reference frame and previous frame centered at (b_{cx}, b_{cy}) with a size of $sw \times sh$. s is a scale factor (s=2 in Fig. 2). In current query frame, an image is cropped at the same position of the previous frame.

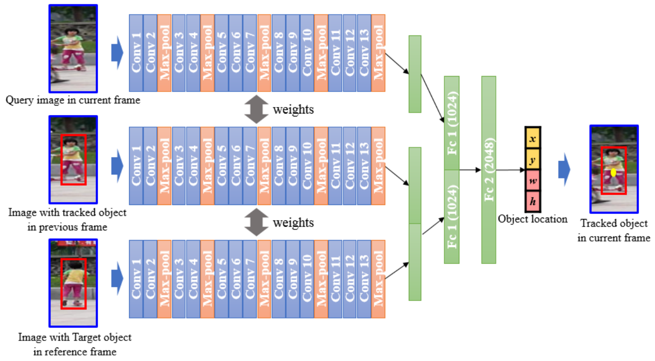


Figure 3: Proposed triplet-CNN architecture for visual tracking.

With three inputs, the proposed triple-CNN predicts the coordinates of the object in current frame. The output is the object bounding box information which consists of the predicted center positions of target object and width and height of the object bounding box. It can be written as $\mathbf{b} = [b_1, b_2, b_3, b_4]$ where b_1=c_x, b_2=c_y, b_3=w, and b_4=h. Fig. 3 shows the architecture of the proposed triplet-CNN. As shown in Fig. 3, the proposed triplet-CNN consists of three individual CNNs and two fully connected layers. For individual CNN, we employ VGG 16-layer network [12]. In every convolutional layer, filters with size of 3x3 are learned followed by rectified linear units. The number of filters in the first convolutional layer is 64. Then, it increases by a factor of 2 in each layer until filter of convolutional layer reaches 512. The max-pooling is conducted with 2 × 2 window and stride 2 after each stage of convolutional layer.

The three individual CNNs share weights to extract features of three inputs in a same feature space. Through the triple-CNN, high-level discriminative features of object are encoded even with different object appearances. The features encoded by CNNs for the tracked object in the previous frame and object in the reference frame are flattened and concatenated. Then nonlinear mapping is performed through the first fully connected layer. The feature encoded by CNN for the current query frame is also flattened and followed by nonlinear mapping using the first fully connected layer. After that, two fully connected layers are concatenated and nonlinear mapping is followed by using second fully connected layer. The output of triplet-CNN directly regresses to object-bounding box for predicting the location of the target object. In the proposed triplet-CNN, by incorporating the previous location of object and different appearance from the reference frame, the location of the target object can be tracked well even in deformed object.

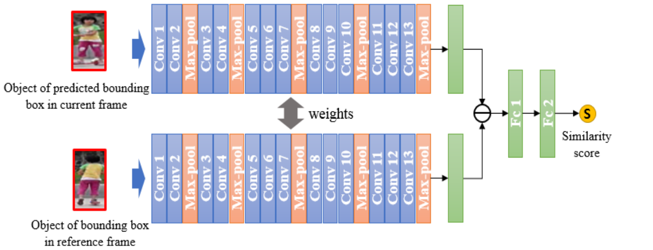


Figure 4: Siamese CNN architecture for measuring the similarity.

For training of the proposed deep network, individual CNN is initialized by the ImageNet pre-trained model and fine-tuned with a visual tracking database.

In the training stage of the proposed triplet-CNN, regression errors are supposed to be minimized. In this paper, the regression loss ($L_{Triplet}$) is calculated using IOU (intersection over union) loss function as follows:

$$L_{Triplet} = -\frac{\sum_{i \epsilon N} G_i * P_i}{\sum_{i \epsilon N} G_i + \sum_{i \epsilon N} P_i - \sum_{i \epsilon N} G_i * P_i}, \quad (1)$$

where G and P are the ground-truth and predicted bounding box. And N is the total number of the pixels. If G_i and P_i are in the ground-truth and predicted bounding box, the value is 1, otherwise 0. IOU loss provides a measurement of the degree of the overlap between ground-truth and predicted bounding box.

2.2 Siamese-CNN based similarity measurement

To prevent drift and wrong tracking by occlusion or deformation, we devise Siamese CNN to verify the quality of predicted bounding box. To measure similarity between the ground-truth and the tracked object, the Siamese CNN takes two inputs; 1) the object of bounding box in reference frame (i.e., ground-truth to be tracked), 2) the object of bounding box predicted by the triplet-CNN. Note that, in the Siamese CNN, surrounding regions of the object are excluded.

Fig. 4. shows the proposed Siamese CNN for measuring the similarity between reference object and predicted object. As shown in the Fig. 4, the proposed Siamese CNN consists of two individual CNNs and two fully connected layers. For two individual CNNs, we employ VGG 16-layer network structure [12] same as the triplet-CNN. Also, two individual CNNs share weights. To effectively learn the similarity between features encoded by CNNs for reference object and predicted object, two outputs of last convolution layer are subtracted.

For training, the proposed Siamese CNN is initialized by the ImageNet pre-trained model and fine-tuned with a visual tracking database. In the training stage, output of the Siamese CNN is set to 0 when target object in current frame is considerably different from the reference, otherwise 1. The proposed Siamese CNN determines whether the two objects (i.e., reference object and predicted object) are similar or not by minimizing similarity classification errors. In this paper, the similarity classification error is defined as binary cross-entropy loss function, which can be

$$L_{Siamese} = -ylog\hat{y} - (1-y)\log(1-\hat{y}), \quad (2)$$

written as

where y is a ground-truth label. $\hat{y}$ is the predicted similarity score between reference object and predicted object by the proposed triplet-CNN.

High similarity score of the Siamese CNN output indicates that both reference and predicted objects of bounding boxes are similar. Therefore, it is highly likely to correctly track target object for the given reference object. On the other hand, low similarity score indicates that the objects in both bounding boxes are different. It means predicted object is falsely tracked due to severe occluded or deformation.

If Siamese CNN output is low similarity score, object tracking in current frame is estimated by considering target object movement. For previous N frames, object bounding box $\mathbf{b}_t$ at the t-th frame can be estimated as

$$\mathbf{b}_t = \frac{\sum_{i=1}^{N}(W_{t-i}\mathbf{b}_{t-i})}{\sum_{i=1}^{N}(W_{t-i})}, \quad (3)$$

where $\mathbf{b}_{t-i}$ is the predicted bounding box at the (t-i)-th frame. W_{t-i} is weight value regarding the $\mathbf{b}_{t-i}$. The weight value regularizes the reliability of the (t-i)-th frame, which can be written as

$$W_{t-i} = \frac{1}{K_t}\exp\left(-\frac{1-\hat{y}_{t-i}}{\sigma_s^2}\right)\exp\left(-\frac{i-1}{\sigma_r^2}\right), \quad (4)$$

where $\hat{y}_{\mathrm{t-i}}$ is the (t-i)-th frame similarity score and K_t is a normalizing parameter to ensure that $\sum_{i=1}^{N}(W_{t-i}) = 1$. The parameters σ_s and σ_t adjust the sensitivity of similarity score and previous frame range, respectively. The first exponential term in (4) gives the reliability of the predicted bounding box at (t-i)-th frame, $\mathbf{b}_{t-i}$. The higher the similarity score is, the more reliable the predicted bounding box is. The second exponential term gives the reliability of previous frames. The closer to the current frame is, the higher the weight value is.

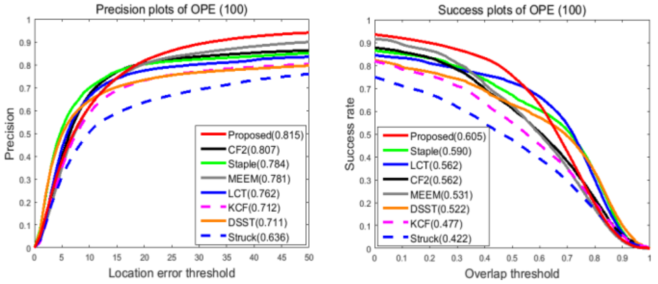
3 EXPERIMENTS

3.1 Experimental Setup

The proposed tracking algorithm was implemented in Python using Keras [13], and ran at about 25 fps on a PC with intel i7 3770 CPU (3.4 GHz) and NVIDIA GeForce GTX 1080. The images of each video were 3-channel image. If the images of video were one channel gray image, we copied the gray image and put it on the 3 channels. The bounding box of the target in the first frame was given. Siamese CNN similarity threshold (Th) was set to 0.3.

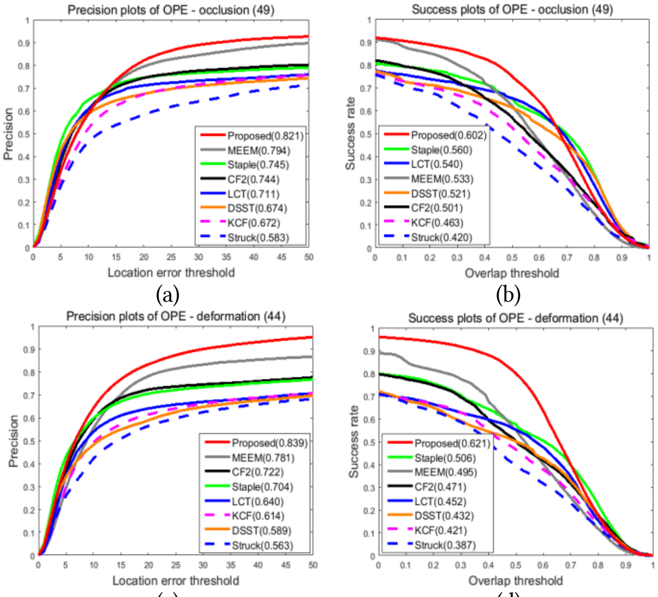
3.2 Datasets

For training triplet-CNN and Siamese CNN, VOT2014 dataset [14] and VOT2015 dataset [15] were used. VOT2014 and VOT2015 have 25 and 60 fully annotated videos, respectively. Both datasets have object variations like occlusion and deformation. For test stage, OTB100 dataset [16] was used. In OTB100 dataset, we used 100 videos which were fully annotated with the information of object variations such as occlusion, deformation, etc. In the dataset, 49 videos contained object occlusion cases and 44 videos contained deformation cases.



(a) (b)

Figure 5: Comparison of tracking performances with 100 videos on OTB100 dataset. (a) Precision plots. The number in the legend in (a) indicates precision score at local error threshold of 20 pixels. (b) Success plots. The number in the legend in (b) indicates AUC.



(a) (b)

(c) (d)

Figure 6: Comparison of tracking performances in the occlusion and deformation cases on OTB100 dataset. (a) and (c) are precision plots for occlusion cases of 49 videos, deformation cases of 44 videos, respectively. (b) and (d) are success plots in the occlusion and deformation cases, respectively.

3.3 Evaluation Metrics

In experiment, the tracking performance was measured by two metrics: center location error and overlap ratio [16]. The center

Table 1: Center location error (pixels) of the proposed method and existing trackers on OTB 100 dataset. The best results and the second best results are marked in red and blue, respectively.

	Ours	MEEM	CF2	Staple	LCT	KCF	DSST	Struck
Coke	9.8	11.8	10.5	23.0	21.6	15.2	21.2	12.1
Girl2	10.1	50.5	117.6	121.2	300.0	263.9	131.5	150.6
Soccer	15.1	65.5	37.6	70.0	63.6	40.0	29.5	82.9
Jump	50.9	124.6	140.0	200.0	174.5	136.8	148.0	143.2
Carscale	15.4	78.1	71.7	31.5	50.7	83.8	40.8	96.8
Freeman4	10.1	28.2	12.3	22.1	12.2	39.3	23.7	48.0
Tiger2	11.7	19.7	19.4	14.2	17.3	46.7	41.6	21.0
Bird1	12.4	121.8	58.2	59.9	102.8	145.0	137.3	148.0
Dudek	12.1	37.9	37.3	68.0	29.6	36.7	25.7	35.9
Couple	15.9	18.4	21.1	30.0	20.0	47.3	113.0	24.4
Skater2	30.1	36.1	38.6	53.6	36.5	42.6	63.0	37.6
Box	22.6	156.4	110.3	59.3	159.9	94.7	101.5	127.0

Table 2: Overlap ratio (%) of the proposed method and existing trackers on OTB 100 dataset. The best results and the second best results are marked in red and blue, respectively.

	Ours	MEEM	CF2	Staple	LCT	KCF	DSST	Struck
Coke	67.2	64.9	64.8	56.0	64.4	56.0	56.5	66.6
Girl2	72.2	53.8	7.1	10.9	6.6	5.7	9.4	22.3
Soccer	57.7	28.0	46.9	21.3	13.3	41.5	43.3	18.2
Jump	30.4	13.4	14.4	5.1	4.6	9.0	8.3	10.2
Carscale	78.5	41.9	42.4	77.6	69.3	42.6	75.1	41.9
Freeman4	63.2	30.5	48.0	41.5	44.3	18.5	37.4	17.1
Tiger2	65.6	52.7	55.7	66.7	61.9	34.2	31.8	53.3
Bird1	49.2	13.0	20.9	28.4	22.2	5.2	10.1	9.0
Dudek	84.3	70.8	72.8	65.3	77.0	72.2	77.3	72.4
Couple	60.5	58.4	55.1	52.4	42.5	20.0	8.4	51.9
Skater2	65.3	60.6	60.7	41.8	58.7	55.1	41.5	57.6
Box	62.8	27.6	28.4	35.5	10.4	30.1	34.3	20.4

location error was defined as Euclidean distance between the center positions of predicted object bounding box and ground-truth bounding box at each frame. The overlap ratio (S) could be written as

$$S = \frac{\text{area}(B_{\text{G}} \cap B_{\text{T}})}{\text{area}(B_{\text{G}} \cup B_{\text{T}})}, \tag{5}$$

where B_{G} and B_{T} indicated ground-truth bounding box and predicted target object bounding box, respectively.

To visualize tracking results measured by center location error and overlap ratio, we used precision plot and the success plot [16]. The precision plot presented the percentage of frames where center location error was within given threshold distance [16]. The representative precision score with 20-pixel distance threshold was employed [16]. The success plot showed the percentage of frames with satisfied $S > t_0$ for all threshold $t_0 \in [0, 1]$. The area under curve (AUC) of success plot was used to quantitatively rank trackers [16].

We illustrated the precision and success plots using the one-pass evaluation (OPE) [16] method. The OPE is a method of measuring tracking performance without re-initialization. To evaluate the performance of tracking in sequences that include occlusion and deformation, we compared our method with 7 existing methods whose speed of the tracker was higher than 15 fps. These were Staple [17], LCT [7], CF2 [11], MEEM [18], DSST [19], KCF [20], and Struck [21].

3.4 Quantitative comparisons

Fig. 5 showed the comparison of tracking performances with 100 videos on OTB100 dataset for the proposed method and existing methods. Note that all the results of the 7 existing methods were provided by the authors. CF2 had the highest precision score among existing methods, which was 0.807. On the other hand, precision score of the proposed method was 0.815 (0.8% gain over the FC2). The highest AUC among existing methods was showed in Staple, which was 0.590. The proposed method achieved 0.605

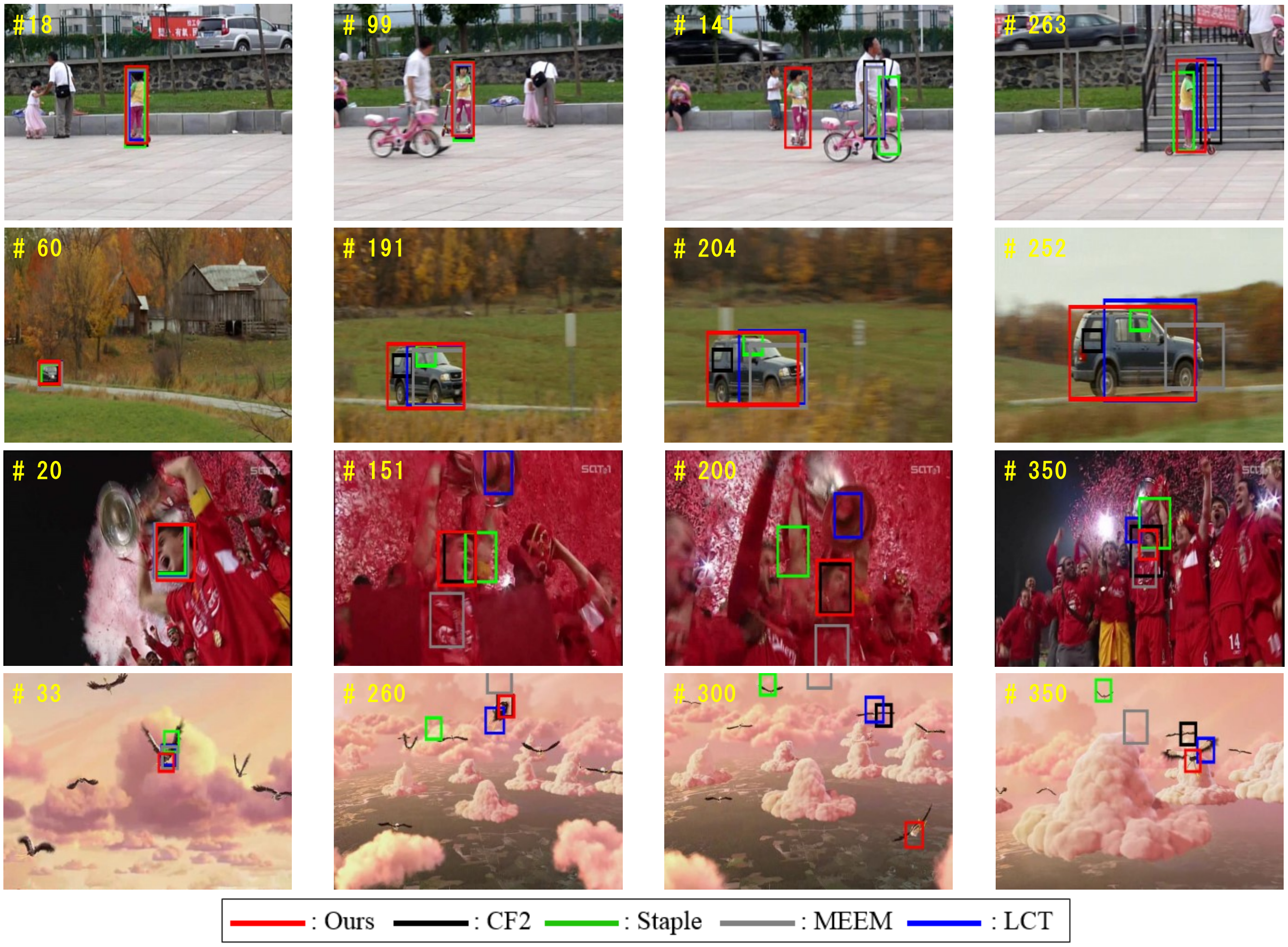


Figure 7: Examples of tracking results at challenging frames. First and second rows are *Girl2* and *Carscale* sequences, respectively. Third and fourth rows are *Soccer* and *Bird1* sequence, respectively. Red bounding box indicates our tracking result. Black and green indicate the CF2 and Staple results, respectively. Gray and blue indicate the MEEM and LCT results, respectively.

AUC in the success plot (1.5% gain over the Staple). To see the robustness of the tracking an object, we compared success plots of Fig. 5 when threshold was 0. As is seen, the proposed method showed highest score in success plots at overlap threshold 0. Moreover, the proposed method showed that target object was tracked well in quite large range of the threshold.

Fig. 6 illustrated the comparison of tracking performances in the occlusion and deformation cases for OTB100 dataset. We used 49 videos for occlusion cases, and 44 videos for deformation cases. As shown in Fig. 6, the proposed method outperformed other methods. In particular, the precision score and AUC of CF2, MEEM, Staple and LCT methods were significantly degraded as compared to the results in Fig. 5 (tracking performances with 100 videos on OTB100 dataset). On the other hand, the proposed method showed robustness against the occlusion and deformation cases. As shown in Fig. 6(a) and 6(b), in the occlusion case, performance of the proposed method was enhanced by 0.6% in precision score and degraded by 0.3% in AUC as compared to the corresponding results in Fig. 5. On the other hand, existing methods had an average performance degradation of 3.9% in precision score and 1.8% in AUC as compared to the corresponding results in Fig. 5. In the deformation case, as shown in the Fig. 6(c) and 6(d), the proposed method was enhanced by 1.8% in precision score and 1.9% in the AUC as compared to the corresponding results in Fig. 5. However, existing methods had an average performance degradation of 7.1% in precision score and 7.2% in AUC score as compared to the corresponding results in Fig. 5. It demonstrated that proposed method outperformed the existing methods in both measures.

To evaluate tracking performance when occlusion and deformation frequently happen, we performed experiments with 12 challenging videos where occlusion and deformation occurred frequently [16]. Table 1 and Table 2 showed the measured center location error and overlap ratio *S*, respectively. Note that smaller center location error and higher overlap ratio mean more accurate tracking results. As shown in the tables, our tracking results showed better tracking performances compared with existing methods.

Occlusion. In [16], *Coke, Girl2, Soccer, Freeman4, Carscale, Box*, and *Tiger2* sequences contained the severe occlusion. As shown in Table 1 and Table 2, in the 6 sequences, all of our center location error are the lowest and most of our overlap ratio show the highest score. The first and second rows of Fig. 7 show *Girl2* and *Carscale* sequence which contains a full occlusion case. After full occlusion, the CF2, Staple, MEEM, and LCT could not consistently track target object (i.e., a girl or car). On the other hand, the proposed method could track after full occlusion because our tracker was supposed to detect this severe occlusion case by using the Siamese-CNN. Then, our tracker predicted target object location based on the previous tracking information. The third row of Fig. 7 showed tracking results in *Soccer* sequence. As shown in the third row of the Fig. 7, despite severe occlusion, the proposed method consistently tracked target object in video sequence. However, CF2, Staple, MEEM, and LCT methods lost the target object. These experimental results demonstrated that the proposed method provided better tracking performances than the existing methods on the severe occlusion case.

Deformation. *Jump, Bird1, Dudek, Couple, and Skater2* sequences contained significant object deformation. As shown in Table 1 and Table 2, the proposed method show small center location errors and higher overlap ratio than other methods. In particular, the LCT, KCF and Struck methods could not deal with deformation case. The fourth rows of Fig. 7 show the tracking results for *Bird1* sequence. For these large object movement and abrupt shape change, existing methods could not track correct target position. On the other hand, the proposed method has correctly tracked in these challenging video sequences.

Experimental results showed that the proposed method could provide outperformed tracking performances robust to severe occlusion and deformation, compared to the existing methods. By cooperating with reference object and previous frame information in learning process, the proposed triplet-CNN could consider the appearance change information of target object. In addition, the proposed Siamese CNN based similarity measurement played important role in preventing the drift or wrong tracking. Consequently, the proposed method could correctly track target object in real-time (about 25 fps) in video sequences containing the challenging environments.

4 CONCLUSIONS

In this paper, we proposed a robust and real time visual tracking in order to handle object occlusion and deformation in video sequences. The proposed deep network for visual tracking was designed to consider appearance changes of target object by using triplet-CNN. In addition, to detect severe occlusion or deformation case, Siamese CNN was devised for similarity measure with reference information. The experimental results showed that the proposed visual tracking method outperformed existing methods.

ACKNOWLEDGEMENT

This work was supported by the ICT R&D program of MSIP/IITP. [2014-0-00077, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis]

REFERENCES

[1] Niu, W., Jiao, L., Han, D., and Wang, Y.F., 2003. Real-time multi person tracking in video surveillance. In *Proceedings of the Pacific Rim Multimedia Conference on* IEEE, 1144-1148.

[2] Aggarwal, J. K., and Ryoo, M. S., 2011. Human activity analysis: a review. *ACM Computing Surveys* 43, 3, 1-43.

[3] Crowley, J. K., and Schewerdt, K., 1999. Robust tracking and compression for video communications. In *Proceedings of the IEEE International Conference Recognition, Analysis and Tracking of Faces and Gestures in Real-Time*, 2-9.

[4] Menresa, C., Varona, J., Mas, R., and Perales, F. J., 2005. Hand tracking and gesture recognition for human-computer interaction. *Electronics Letters on Computer Vision and Image Analysis 4*, 3, 96-104.

[5] Adam, A., Rivlin, E., and Shimshoni, I., 2006. Robust fragments-based tracking using the integral histogram. In *Proceedings of the International Conference on Computer Vision*, 798-805.

[6] Shu, G., Dehghan, A., Oreifeg, O, Hand, E., and Shah, M., 2012. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* IEEE, 1815-1821.

[7] Ma, C., Yang, X., Zhang, C., and Yang, M. H., 2015, Long-term correlation tracking. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* IEEE, 5388-5396.

[8] Wang, L., Ouyang, W., Wang, X., Lu, H., 2015. Visual tracking with fully convolutional networks, In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* IEEE, 3119-3127.

[9] Nam, H., Han, B., 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* IEEE, 4293-4302.

[10] Held, D., Thrun, S., Savarese, S., 2016. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, 749-765.

[11] Ma, C., Huang, J. B., Yang, X., and Yang, M. H., 2015. Hierarchical convolutional feature for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 3074-3082.

[12] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv, 1409.1556.*

[13] Chollet, F., 2015. Keras. Available: https://github.com/fchollet/keras

[14] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., et al., 2014. In *European Conference on Computer Vision Workshop*, 191-217.

[15] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., et al., 2015. The visual object tracking vot2015 results. In *Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW)*, 1-23.

[16] Wu, Y., Lim, J., Yang, M. H., 2015. Object tracking benchmark. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 37*, 9, 1834-1848.

[17] Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. H., 2016. Staple: complementary learners for real-time tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* IEEE, 1401-1409.

[18] Zhang, J., Ma, S., Sclaroff, S., 2014. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, 188-203.

[19] Danelljan, M., Hager, G. K., and Felsberg, M., 2014. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham*, 1-11.

[20] Henriques, J. F., Caseiro, R., Martins, P., and Bastista, J., 2015. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 37*, 3, 583-896.

[21] Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. H. S., 2016. Staple: Complementary learners for real-time tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* IEEE, 1401-1409.