creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:47020-200000106373

Master's Thesis

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

Mooyeol Baek (백 무 열)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2018

dCollection @ postech

트리 구조의 다중 컨볼루션 신경망을 이용한 물체 추적

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

by

Mooyeol Baek
Department of Computer Science and Engineering
Pohang University of Science and Technology

A thesis submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science and Engineering.

Pohang, Korea
June 12, 2018
Approved by

Suha Kwak

Academic Advisor

Modeling and Propagating CNNs in a Tree Structure for Visual Tracking

Mooyeol Baek

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a Master's degree from POSTECH.

June 12, 2018

Committee Chair Suha Kwak

Member Bohyung Han

Member Minsu Cho

MCSE 백 무 열, Mooyeol Baek,
20142675 Modeling and Propagating CNNs in a Tree Structure for Visual Tracking,
트리 구조의 다중 컨볼루션 신경망을 이용한 물체 추적,
Department of Computer Science and Engineering, 2018, 37P,
Advisor: Suha Kwak. Text in English.

ABSTRACT

This work is an online visual tracking algorithm managing multiple target appearance models stochastically in a tree structure. The proposed algorithm employs Convolutional Neural Networks (CNNs) to represent multiple target appearances, where multiple CNNs collaborate to estimate target states and determine the desirable paths for online model updates in the tree. The strategy maintaining multiple CNNs in diverse branches of tree structure enables to deal with multi-modality in target appearances and preserve model reliability through smooth updates along tree paths. Since multiple CNNs share all parameters in convolutional layers, it takes advantage of multiple models with little extra cost by saving memory space and avoiding redundant network evaluations. The final target state is estimated by sampling target candidates around the state in the previous frame and identifying the best sample in terms of a weighted average score from a set of active CNNs. Proposed algorithm achieved outstanding performance compared to the state-of-the-art techniques in challenging datasets such as online tracking benchmark and visual object tracking challenge.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교


POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

Contents

1 Introduction **1**

2 Related Work **4**

2.1 Visual Trackers 4

2.2 CNN for Visual Tracking 5

2.3 Visual Tracking with Multiple Models 6

3 Tree-Structured CNNs (TCNN) for Visual Tracking **7**

3.1 Algorithm Overview 7

3.2 Tree Construction 9

3.3 CNN Architecture 10

4 Online Tracking Procedure **11**

4.1 Target State Estimation using Multiple CNNs 11

4.2 Bounding Box Regression 13

4.3 Model Update 13

4.4 Implementation Details 14

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

5 Experiment **17**

5.1 Evaluation on OTB dataset 17

5.2 Internal Analysis 25

5.3 Evaluation on VOT2016 Dataset 26

5.4 Evaluation on TempleColor & UAV Datasets 30

6 Conclusion **31**

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

List of Figures

3.1 Illustration of target state estimation and model update procedures using multiple CNNs maintained in a tree structure. The width of a black arrow indicates the weight of a CNN for target state estimation while the width of a red edge denotes the affinity between two CNNs. The width of box outline means the reliability of the CNN associated with the box. 8

3.2 Illustration of the online tracking procedure over time. 9

3.3 The architecture of the proposed network. VGG-M network pretrained on ImageNet is used for convolutional layers while all the fully connected layers are initialized randomly. The number of channels and the size of each feature map are shown with the name of each layer. 10

5.1 Quantitative results on OTB100 [1] . The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively. 18

5.2 Quantitative results for 8 challenge attributes 20

5.3 Plots of TRE and SRE on OTB100 [1]. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively. 22

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

5.4 Qualitative results in several challenging sequences of OTB100 dataset (*Dog1*, *Bolt2*, *Car24*, *Diving*, *Freeman4*, *Girl2*, *Human9*, *Soccer*, and *Trans*). 23
5.5 Failure cases of TCNN on OTB100 dataset (*CarScale*, *Bolt2*, *Biker*, *Human4*). 24
5.6 Quantitative results on VOT2016 [2] with respect to the challenging visual attributes. Trackers on the right side of the graph are better. 28
5.7 Quantitative results for 6 challenge attributes. Trackers on the upper-right side of the graph are better. 29
5.8 Quantitative results on TempleColor [3] dataset. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively. 30
5.9 Quantitative results on UAV [3] dataset. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively. 30

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

List of Algorithms

1 Overall algorithm of TCNN tracker 16

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

List of Tables

5.1 Ablation studies. The rates of successfully tracked frames in terms of center location error with threshold 20 pixels and bounding box overlap ratio with threshold 0.5 are denoted by precision and success, respectively. AUC is based on the success plot in OTB evaluation protocol [1]. 25

5.2 AUC scores on OTB100 of TCNN with varying K and Δ. 25

5.3 Expected average overlap score and the average scores and ranks of accuracy and robustness on the experiment in VOT2016 [2]. These are results on two separate settings for VOT2016, the *baseline* setting uses re-initialize protocol while the unsupervised setting does not. The first and second best scores are highlighted in red and blue colors, respectively. 26

5.4 Expected average overlap score on the experiment for challenging attributes of VOT2016 [2]. The first and second best scores are highlighted in red and blue colors, respectively. 28

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

1 Introduction

Visual tracking is a fundamental task in computer vision with a wide range of real world applications. Among many subproblems in visual tracking, target appearance modeling is one of the most critical components and there are various approaches to maintain adaptive and robust models. This problem is typically formulated in an online learning framework, where a generative or discriminative model is incrementally updated during tracking to adapt the variations in target appearances. Most existing tracking algorithms update target models by assuming that target appearances change smoothly over time. However, this strategy may not be appropriate for handling more challenging situations such as occlusion, illumination variation, abrupt motion and deformation, which may break temporal smoothness assumption. Some algorithms employ multiple models [4–6], multi-modal representations [7] or nonlinear classifiers [8, 9] to address these issues. However, the constructed models are still not strong enough and online model updates are limited to sequential learning in a temporal order, which may not be able to make the models sufficiently discriminative and diverse.

CNNs have recently gained significant attention in various visual recognition problems such as image classification [10], object detection and localization [11], image segmentation [12, 13] due to their strong representation power. However, online learning with CNNs is not straightforward because neural networks tend to forget previously

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

learned information quickly when they learn new information [14]. This property often incurs drift problem especially when background information contaminates target appearance models, targets are completely occluded by other objects, or tracking fails temporarily. This problem can be alleviated by maintaining multiple versions of target appearance models constructed at different time steps and updating a subset of models selectively to keep a history of target appearances. This idea has been investigated in [15], where a pool of CNNs are used to model target appearances, but it does not consider the reliability of each CNN to estimate target states and update models.

This work is an online visual tracking algorithm, which estimates target states using the likelihoods obtained from multiple CNNs. The CNNs are maintained in a tree structure and updated online along the path in the tree. The proposed tree structure is *stochastic* since the construction of new nodes is determined probabilistically depending on classification scores given by CNNs stored in the tree. Since each path keeps track of a separate history about target appearance changes, the proposed algorithm is effective to handle multi-modal target appearances and other exceptions such as short-term occlusions and tracking failures. In addition, since the new model corresponding to the current frame is constructed by fine-tuning the CNN that produces the highest likelihood for target state estimation, more consistent and reliable models are to be generated through online learning only with few training examples. The main contributions of this thesis are summarized below:

- Proposed visual tracking algorithm manages target appearance models based on CNNs in a tree structure, where the models are updated online along the path in the tree. This strategy enables tracker to learn more persistent models through smooth updates.

- The proposed tracking algorithm utilize multiple models, which are constructed stochastically, to capture diverse target appearances and performs more robust tracking even with challenges such as appearance changes, occlusions, and tem-

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

porary tracking failures.

- The proposed algorithm presents outstanding accuracy in the standard tracking benchmarks [1, 2] and outperforms the state-of-the-art methods.

The rest of this paper is organized as follows. First, Chapter 2 reviews various techniques in visual tracking in , and then the structure of the proposed approach is described in Chapter 3. Chapter 4 discusses the detailed visual tracking procedure and Chapter 5 illustrates experimental results in multiple challenging datasets.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

2 Related Work

2.1 Visual Trackers

There are various kinds of visual tracking algorithms, and it is difficult to review all the prior works. This section focuses on several discriminative tracking algorithms based on tracking-by-detection first, and then discuss correlation-filter-based algorithms which are emerging lately.

Tracking-by-detection approaches formulate visual tracking as a discriminative object classification problem in a sequence of video frames. The techniques in this category typically learn classifiers to differentiate targets from surrounding backgrounds; various algorithms have achieved improved performance by coping with dynamic appearance changes and constructing robust target models. For example, [8] modified a famous object detection algorithm, Adaboost, and presented an online learning method for tracking. A multiple instance learning technique has been introduced in [16] to update classifier online, where a bag of image patches is employed as a training example instead of a single patch to alleviate labeling noises. By similar motivation, an approach based on structured SVM has been proposed in [9]. TLD [17] proposed a semi-supervised learning technique with structural constraints. All of these techniques are successful in learning reasonable target representations by adopting online discriminative learning procedures,

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

but still rely on simple shallow features; tracking performance may be improved further by using deep features.

These days, visual trackers with correlation filter have drawn attention due to their efficiency and accurate tracking result. MOSSE [18] proposed an efficient way to generate correlation filters which are robust to appearance changes and are better at discriminating between targets and background. Henriques *et al.* [19] formulated kernelized correlation filters (KCF) using circulant matrices, and efficiently incorporated multi-channel features in a Fourier domain. Several variations of KCF tracker have been subsequently investigated to improve tracking performance. For example, DSST [20] learns separate filters for translation and scaling, and MUSTer [21] employs short-term and long-term memory stores inspired by a psychological memory model. Although these approaches are satisfactory in constrained environments, they have an inherent limitation that they resort to low-level hand-crafted features, which are vulnerable to dynamic situations including illumination changes, occlusion, deformations, background clutter, etc.

2.2 CNN for Visual Tracking

Although the representations by deep neural networks turn out to be effective in various visual recognition problems, tracking algorithms based on hand-crafted features [4, 21] often outperform CNN-based approaches. This is partly because CNNs are difficult to train using noisy labels online while they are prone to overfit to a small number of training examples; it is not straightforward to apply CNNs to visual tracking problems involving online learning. For example, the performance of [15], which is based on shallow custom neural networks, is not as successful as recent tracking methods based on shallow feature learning.

Most CNN-based tracking algorithms bypass these challenges using the CNNs with fixed parameters transferred from a model pretrained on ImageNet [22]. For example, simple approaches based on fully convolutional networks [23] or hierarchical repre-

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

sentations [24] present substantially improved results. In addition, the combination of pretrained CNN and online SVM achieves competitive results [25]. Correlation filter with pretrained CNN features [26, 27] is one of the state-of-the-art techniques. These CNN-based trackers started to present competitive accuracy in the online tracking benchmark [1].

On the other hand, stochastic learning is a useful idea to regularize CNN with small and noisy training data. Dropout is a popular regularization scheme by randomly turning off the activations in fully-connected layers. Furthermore, [28] proposes a concept of stochastic depth, which randomly turns off the layers and feed the intermediate input to the next layer. Proposed tracker utilizes the success of these works, and borrowed the concept of stochastic learning to the update scheme.

2.3 Visual Tracking with Multiple Models

Multiple models are often employed in generative tracking algorithms to handle target appearance variations and recover from tracking failures. Trackers based on sparse representation [5, 6] maintain multiple target templates to compute the likelihood of each sample by minimizing its reconstruction error while [29] integrates multiple observation models via an MCMC framework. Nam *et al*. [30] integrate patch-matching results from multiple frames and estimate the posterior of target state. On the other hand, ensemble classifiers have sometimes been applied to visual tracking problem. Tang *et al*. [31] proposed a co-tracking framework based on two support vector machines. An ensemble of weak classifiers is employed to estimate target states in [32, 33]. Zhang *et al*. [4] presented a framework based on multiple snapshots of SVM-based trackers to recover from tracking failures.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

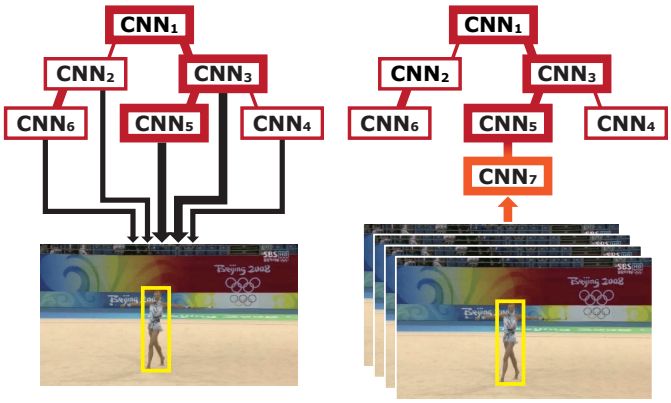
3 Tree-Structured CNNs (TCNN) for Visual Tracking

3.1 Algorithm Overview

The proposed algorithm maintains multiple target appearance models based on CNNs in a stochastic tree structure to preserve model consistency and handle appearance multi-modality effectively. The proposed approach consists of two main components as in ordinary tracking algorithms—state estimation and model update—whose procedures are illustrated in Figure 3.1. Note that both components require interaction between multiple CNNs. When a new frame is given, the tracker draw candidate samples around the target state estimated in the previous frame, and compute the likelihood of each sample based on the weighted average of the scores from multiple CNNs. The weight of each CNN is determined by the reliability of the path along which the CNN has been updated in the tree structure. The target state in the current frame is estimated by finding the candidate with the maximum likelihood.

After tracking a predefined number of frames, a CNN in the tree-structured model is updated by one of the two update strategies. One is gradual update, which updates one of the existing CNNs with the highest weight for target state estimation. The other is drastic update, where a new node is constructed and fine-tuned extensively with more iterations. The gradual update is helpful to ensure smooth model changes while the drastic update

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교



(a) State estimation (b) Model update

Figure 3.1: Illustration of target state estimation and model update procedures using multiple CNNs maintained in a tree structure. The width of a black arrow indicates the weight of a CNN for target state estimation while the width of a red edge denotes the affinity between two CNNs. The width of box outline means the reliability of the CNN associated with the box.

is effective to maintain multi-modal appearances of target. On each update, one of the two update strategies is selected randomly based on a Bernoulli distribution with adaptive parameters; drastic update is more likely to be selected when the estimated target score has been low in recent frames. Due to this stochastic nature, the tracker reduce unnecessary node construction while creating a new model if the confidence of existing models is consistently low over time. This two-track strategy is useful to maintain multi-modality without loss of reliability; the stochastic update scheme diversifies models and prevents both overfitting and underfitting in practice.

This approach has something in common with [15] and [34] because they employ CNN as a binary classifier. However, [34] uses a single CNN pre-trained on tracking datasets to capture common knowledge discriminating target and background. Multiple CNNs are used in [15], but this method selects k nearest CNNs based on prototype matching distances for tracking. Proposed algorithm is differentiated from these approaches since it is more interested in how to keep multi-modality of multiple CNNs

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

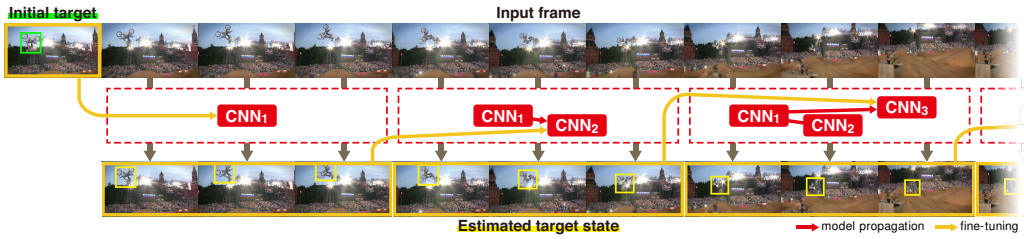


Figure 3.2: Illustration of the online tracking procedure over time.

and maximize their reliability for tracking by introducing a novel model maintenance technique using a stochastic tree structure. Visual tracking based on a tree-structured graphical model has been investigated in [35], but this work is focused on identifying the optimal density propagation path for offline tracking. The idea in [30] is also related, but it mainly discusses posterior propagation on directed acyclic graphs for visual tracking.

3.2 Tree Construction

The proposed tracker maintain a tree structure to manage hierarchical multiple target appearance models based on CNNs. In the tree structure $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$, a vertex $v \in \mathcal{V}$ corresponds to a CNN and a directed edge $(u, v) \in \mathcal{E}$ defines the relationship between CNNs. The score of an edge (u, v) is the affinity between two end vertices, which is given by

$$s(u, v) = \frac{1}{|\mathcal{F}_v|} \sum_{t \in \mathcal{F}_v} \phi_u(\mathbf{x}_t^*), \tag{3.1}$$

where $\mathcal{F}_v$ is a set of consecutive frames that is used to train the CNN associated with v, $\mathbf{x}_t^*$ is the estimated target state at frame t, and $\phi_u(\cdot)$ is the predicted positive score with respect to the CNN in u. Note that the edge scores play crucial roles in estimating target states and providing reliable paths for model update. The details about this issue are discussed next.

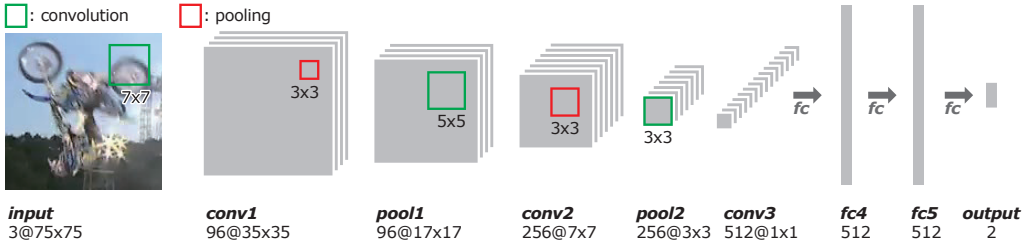


Figure 3.3: The architecture of the proposed network. VGG-M network pretrained on ImageNet is used for convolutional layers while all the fully connected layers are initialized randomly. The number of channels and the size of each feature map are shown with the name of each layer.

3.3 CNN Architecture

Tracker network consists of three convolutional layers and three fully connected layers. The convolution filters are identical to the ones in VGG-M network [36] pretrained on ImageNet [22]. The last fully connected layer has 2 units for binary classification while the preceding two fully connected layers are composed of 512 units. All weights in these three layers are initialized randomly. The input to the network is a 75×75 RGB image and its size is equivalent to the receptive field size of the only single unit (per channel) in the last convolutional layer. Note that, although this tracker borrow the convolution filters from VGG-M network, the size of network is smaller than the original VGG-M network. The output of an input image $\mathbf{x}$ is a normalized vector $[\phi(\mathbf{x}), 1 - \phi(\mathbf{x})]^{\mathrm{T}}$, whose elements represent scores for target and background, respectively. The overall architecture of tracker CNN is illustrated in Figure 3.3.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

4 Online Tracking Procedure

This chapter describes the architecture of the CNN employed to learn discriminative target appearance models. Then, the detailed procedure of the proposed tracking algorithm follows, which includes the method to maintain multiple CNNs in a tree structure for robust appearance modeling.

4.1 Target State Estimation using Multiple CNNs

The proposed algorithm estimates the target state in a new frame by aggregating the scores from multiple CNNs in the tree structure. Let $\mathbf{x}_t^1, \dots, \mathbf{x}_t^N$ be N target candidates in the current frame t, which are sampled around the previous target state, and $\mathcal{V}_+ \subseteq \mathcal{V}$ be the active set of CNNs, which contribute to the state estimation. $\mathcal{V}_+$ contains $K(=10)$ CNNs, which are the ones added to the tree structure most recently. The optimal target state $\mathbf{x}_t^*$ is determined among the candidate samples with the maximum target score, which is computed by a weighted average of CNN scores. In other words, $\mathbf{x}_t^*$ is given by

$$\mathbf{x}_t^* = \underset{\mathbf{x}_t^i}{\operatorname{argmax}} \sum_{v \in \mathcal{V}_+} w_{v \to t} \phi_v(\mathbf{x}_t^i). \tag{4.1}$$

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

where $w_{v \to t}$ denotes the weight of the CNN corresponding to vertex v for tracking target in frame t. The remaining issue is how to determine the weight $w_{v \to t}$ in Eq. (4.1). The weight is identified by the following two factors: affinity to the current frame and reliability of CNN. The affinity $\alpha_{v \to t}$ indicates how confidently a CNN in vertex v affects the tracking result in frame t, which is determined by the maximum positive score over all candidates as follows:

$$\alpha_{v \to t} = \max_{\mathbf{x}_t^i} \phi_v(\mathbf{x}_t^i). \tag{4.2}$$

However, the estimation of the weight $w_{v \to t}$ based only on this measure may be problematic because it ignores how reliable each CNN is. For example, if a CNN is fine-tuned on the frames with complete failures or nontrivial errors, the CNN is likely to produce high scores for background objects and should be penalized despite the existence of samples with high affinities.[1]

To address this issue, the tracker also employ the reliability of each CNN for target state estimation using the path in the tree structure along which the CNN has been updated. Note that the tree may have an unreliable CNN in the path, which may has been updated using the frames with tracking errors. The reliability of CNN associated with vertex v is estimated, which is denoted by β_v, using the score in the bottleneck edge. Specifically, the reliability of a CNN is efficiently computed in a recursive manner without exploring the entire path, which is formally given by

$$\beta_{root} = 1, \text{ otherwise } \beta_v = \min\left(s(p_v, v), \beta_{p_v}\right), \tag{4.3}$$

where p_v is the parent of CNN in vertex v, which has an outgoing edge (p_v, v) in the tree structure.

Based on these two criteria, the combined weight of the CNN corresponding to ver-

[1] Such CNNs are not completely useless since sometimes following background objects helps tracking targets in case of severe occlusion.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

tex v, $w_{v \to t}$, is given by

$$w_{v \to t} = \min(\alpha_{v \to t}, \beta_v). \tag{4.4}$$

Note that both $\alpha_{v \to t}$ and β_v are the scores evaluated on CNNs, and taking the minimum of two bottleneck similarities determines the weight of a child CNN associated with frame t when it is updated from CNN in vertex v.

4.2 Bounding Box Regression

The target state estimated in Eq. (4.1) may not correspond to the tight bounding box since the representation by CNN is not appropriate for accurate localization of an object. Therefore, the tracker employs the bounding box regression technique, which is frequently used in object detection [11], to enhance target localization quality. The tracker learns a simple regression function at the first frame, and adjust target bounding boxes using the model in the subsequent frames. This idea is simple but effective to improve performance. Please refer to [11] for details.

4.3 Model Update

The proposed tracker maintain multiple CNNs in a tree structure, where a CNN is stored in each node. This approach improves multi-modality of appearance models by having CNNs in diverse branches while it preserves model reliability by smoothly updating CNNs along the path in the tree. Now, the critical issue is how to select the appropriate path for fine-tuning a CNN using new training examples.

Let z be a potential new vertex after finishing tracking $\Delta(= 10)$ consecutive frames without model updates, which are elements in $\mathcal{F}_z$. A vertex $v^* \in \mathcal{V}_+$ is chosen to maxi-

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

mize the reliability of z, identified by

$$v^* = \arg\max_{v \in \mathcal{V}_+} \min(\tilde{s}(v, z), \beta_v), \tag{4.5}$$

where $\tilde{s}(v, z)$ denotes the score of a tentative edge (v, z), which is computed by Eq. (3.1). The algorithm determines probabilistically whether to update gradually or drastically. A binary random variable r_z is obtained by

$$r_z \sim \text{Bernoulli}(\tilde{s}(v^*, z)), \tag{4.6}$$

and the tracker is updated by the gradual model if $r_z = 1$ or by the drastic update otherwise. The gradual update scheme does not create z, and fine-tunes v^* using the training examples collected from two sets of frames, $\mathcal{F}_z \cup \mathcal{F}_{v^*}$. The drastic update adds a new vertex z as a descendant of v^*, and the CNN in vertex z is fine-tuned extensively with more iterations from the CNN in v^* using $\mathcal{F}_z \cup \mathcal{F}_{v^*}$. The active CNN set denoted by $\mathcal{V}_+$ is also updated to maintain maximum $K(= 10)$ CNNs, by adding z and removing the oldest vertex in $\mathcal{V}_+$.

4.4 Implementation Details

To estimate the target state in each frame, the tracker draws 256 samples in (x, y, s) space from an independent multivariate normal distribution centered at the previous target state. The standard deviations of the three variables are given by $\sigma_x = \sigma_y = 0.3l$ and $\sigma_s = 0.5$; l is the average of width and height of the previous target bounding box, and the scale of a sample bounding box is determined by multiplying 1.05^s to the width and height of the initial target.

The tracker only updates the fully connected layers while all the convolutional layers are identical to the original pretrained network on ImageNet. Although the tracker store

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

multiple CNNs in the tree structure, the parameters of all convolutional layers are shared and the outputs from the shared layers can be reused. Therefore, maintaining and evaluating multiple CNNs require little additional cost compared to handling a single CNN in terms of time and space complexity.

The training data are collected whenever tracking is completed in each frame, where the tracker draws 50 positive and 200 negative examples that have larger than 0.7 IoU and less than 0.5 IoU with the estimated target bounding boxes, respectively. In practice, the tracker does not store image patches as training examples but save their conv_3 features since the features in the layer do not change and it is not necessary to perform convolution operations more than once.

CNNs are trained by the standard Stochastic Gradient Descent (SGD) method with softmax cross-entropy loss. The CNNs with the gradual update are trained for 10 iterations with learning rate 0.003, while parameters for the drastic update are 50 iterations with learning rate 0.003. The initial CNN is trained for 50 iterations with learning rate 0.001. The weight decay and momentum parameters are given by 0.0005 and 0.9, respectively. All parameters are fixed throughout the experiment. Please refer Algorithm 1 for pseudo-code of overall tracking procedure.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

Algorithm 1: Overall algorithm of TCNN tracker

Data: Sequence of images, Initial target state $\mathbf{x}_1$
Result: Estimated target states $\{\mathbf{x}_t^* | 1 \leq t \leq N\}$

Random initialize the fully-connected layer of ϕ_1.
Train a bounding box regressor.
Sample image patches from the first frame and update ϕ_1.
$t = 0, n = 0, \mathcal{V} \leftarrow \phi_1, \mathcal{V}^+ \leftarrow \phi_1$
while *not at end of video* **do**
    $t \leftarrow t + 1, n \leftarrow n + 1$
    Draw target candidates $\{\mathbf{x}_t^i\}$.
    Find optimal target state $\mathbf{x}_t^*$ by Eq. 4.1.
    Sample image patches from this frame.
    Adjust $\mathbf{x}_t^*$ using bounding box regressor.
    if $n \geq \Delta$ **then**
        Find a parent of the new node v^* by Eq. 4.5
        $\phi_z \leftarrow \phi_{v^*}$ and update ϕ_z using sampled patches.
        Get a random variable r_z by Eq. 4.6
        if $r_z = 1$ **then**
            $\mathcal{V} \leftarrow (\mathcal{V} - \phi_{v^*}) \cup \phi_z$
            $\mathcal{V}^+ \leftarrow (\mathcal{V}^+ - \phi_{v^*}) \cup \phi_z$ `(Gradual update)`
        else
            $\mathcal{V} \leftarrow \mathcal{V} \cup \phi_z$
            $\mathcal{V}^+ \leftarrow \mathcal{V}^+ \cup \phi_z$ `(Drastic update)`
            if $|\mathcal{V}^+| \geq K$ **then**
                $\mathcal{V}^+ \leftarrow \mathcal{V}^+ - \phi_{\text{oldest}}$
        $n \leftarrow 0$

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

5 Experiment

The proposed algorithm is implemented in MATLAB using MatConvNet library [37]. The average speed is approximately 1.5 fps without optimization using an Intel Core i7-5820K CPU with 3.30GHz and a single NVIDIA GeForce GTX TITAN X GPU. The proposed algorithm is tested on four standard datasets for tracking performance evaluation; online tracking benchmark (OTB) [1], visual object tracking 2016 benchmark (VOT2016) [2], templecolor dataset [3], and UAV123 benchmark [38].

5.1 Evaluation on OTB dataset

The evaluation results of proposed algorithm on online tracking benchmark [1] is described on this section. OTB100 contains a hundred fully-annotated video sequences with various targets and backgrounds. The evaluation follows the protocol provided by the benchmark [1], where the performance of trackers is evaluated based on two criteria: bounding box overlap ratio and center location error. The success and precision plots are generated by computing the rates of successfully tracked frames at many different thresholds in the two criteria. The ranks of trackers are determined by the accuracy at 20 pixel threshold in the precision plot and the Area Under Curve (AUC) score in the success plot.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

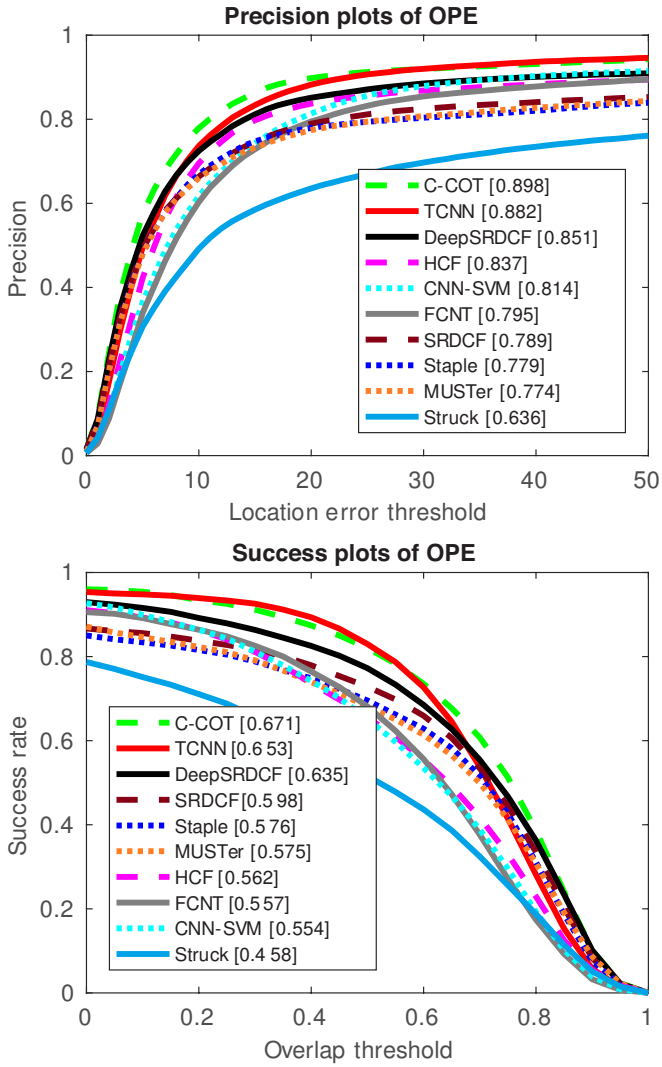


Figure 5.1: Quantitative results on OTB100 [1] . The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively.

The proposed algorithm is compared with the nine state-of-the-art trackers including HCF [24], CNN-SVM [25], FCNT [23], C-COT [27], DeepSRDCF [26], SRDCF [39], Staple [40], MUSTer [21], and Struck [9]. The first five algorithms employ the feature descriptors from CNNs while the rest of the methods are based on the hand-crafted features. The precision and success plots in OTB100 [1] are presented in Figure 5.1. The proposed tracker is denoted by TCNN, which represents the characteristics of the algorithm, observation using tree of CNNs.

The results on the OTB dataset show that TCNN outperforms the most of the trackers. These good results are partly attributed to the strength of CNN features; the learned representations by CNNs are more effective to capture semantic information of a target than low-level hand-crafted features. In particular, TCNN outperforms other CNN-based trackers, which incorporate even deeper networks like AlexNet [10] or VGG-net [41]. The better accuracy of TCNN implies that the proposed model update and state estimation strategy using multiple CNNs based on a tree structure is helpful to deal with various challenges. I also believe that this is because the high-level features obtained from CNNs may contain insufficient spatial information and this is aggravated as the network goes deeper. On the other hand, TCNN is slightly less accurate in the range for strict thresholds compared with C-COT, based on correlation filters on multiple intermediate layers of CNN. It is because employing features from fully-connected layer of CNN are often not as effective as low-level features for the purpose of tight localization.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

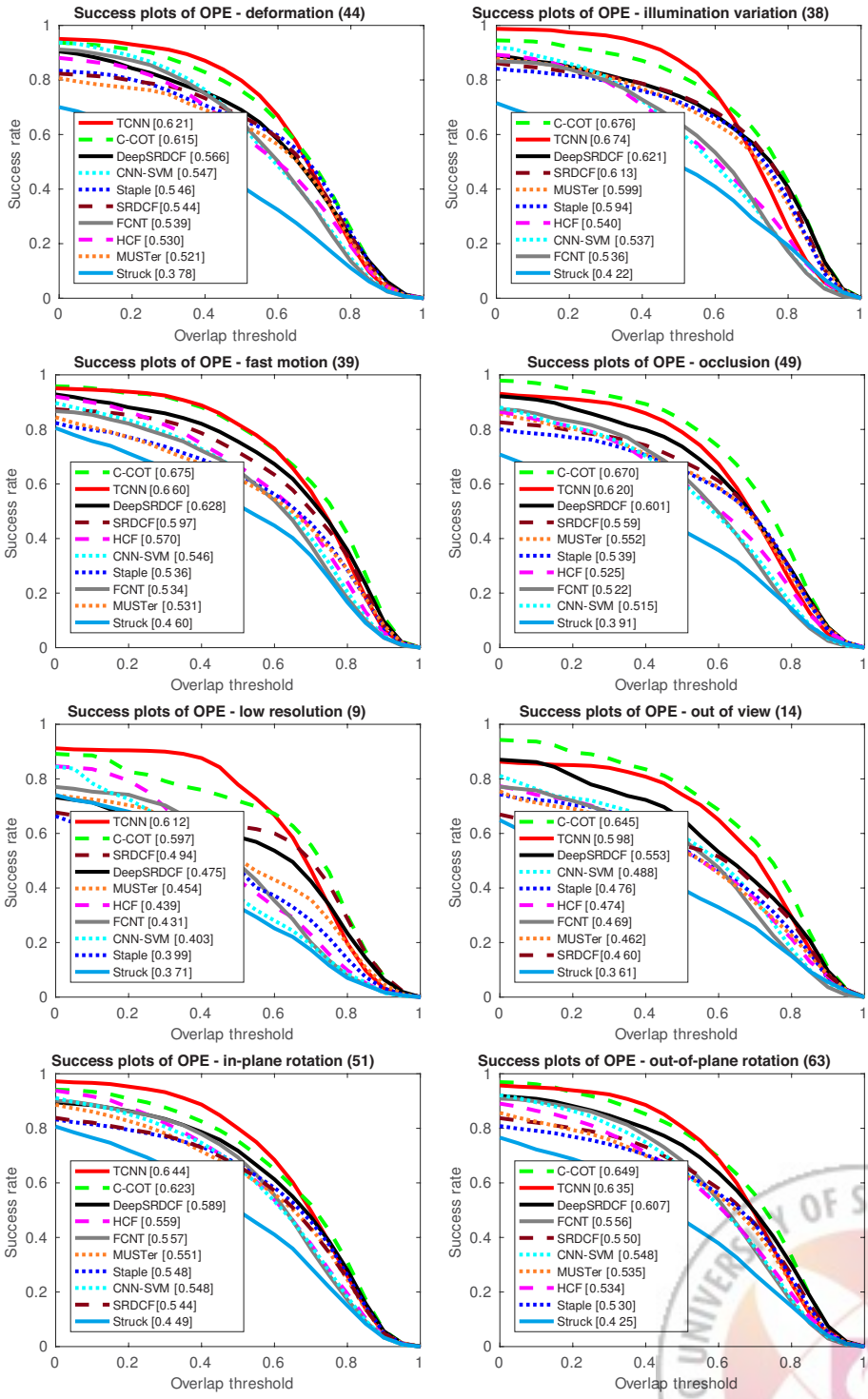


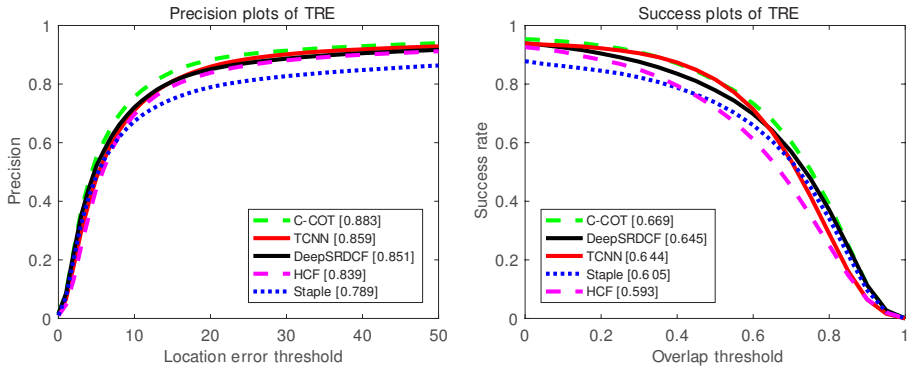
Figure 5.2: Quantitative results for 8 challenge attributes

Figure 5.2 presents the performance of tracking algorithms for various challenging attributes provided in the benchmark; TCNN is effective in handling all kinds of attributes compared to the existing state-of-the-art methods. The trackers based on low-level features generally fail to track targets in challenging situations such as deformation and motion blur, which require high-level understanding about targets. The approaches based on CNN classifiers work well in general, but are not successful typically in illumination variations probably due to dramatic changes in target appearances. TCNN is not outstanding in dealing with out-of-view situation, because it is based on local candidate sampling and does not incorporate any re-detection module. The qualitative tracking results by multiple algorithms on a subset of sequences are illustrated in Figure 5.4, 5.5.

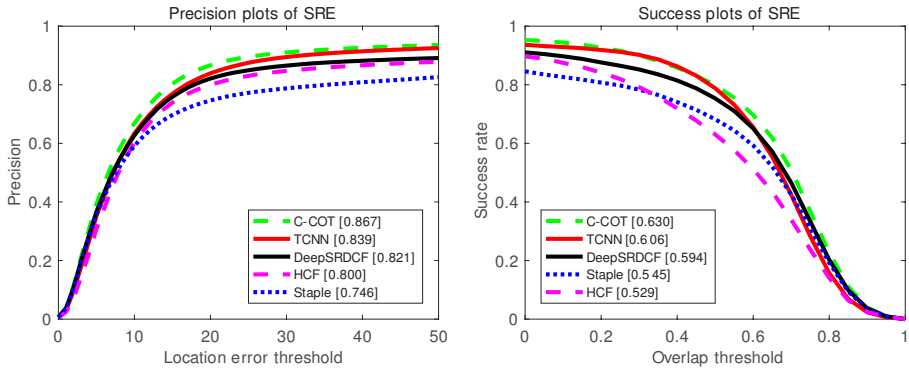
On the other hand, TCNN is also compared on OTB100 dataset with the other evaluation methods, temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) (Figure 5.3). These additional evaluations are designed by the benchmark [1] to measure the sensitivity of the trackers with respect to initialization. TRE measures the performance on the sequences with different start frames, and SRE spatially perturbs the initial bounding boxes. Please refer [1] for more details.

TCNN is compared on OTB100 with four trackers, a subset of the comparing trackers providing the TRE and SRE results. The performances of the trackers are in the same order; TCNN shows the comparable scores with C-COT and outperforms the other trackers. TCNN is less impressive on TRE since TRE tests the challenging situations before the tree-structured model matures. The result shows that the performance of the state-of-the-art trackers on OTB100 is not much sensitive to the perturbation, and the dataset is quite saturated.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교



(a) TRE Results



(b) SRE Results

Figure 5.3: Plots of TRE and SRE on OTB100 [1]. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively.

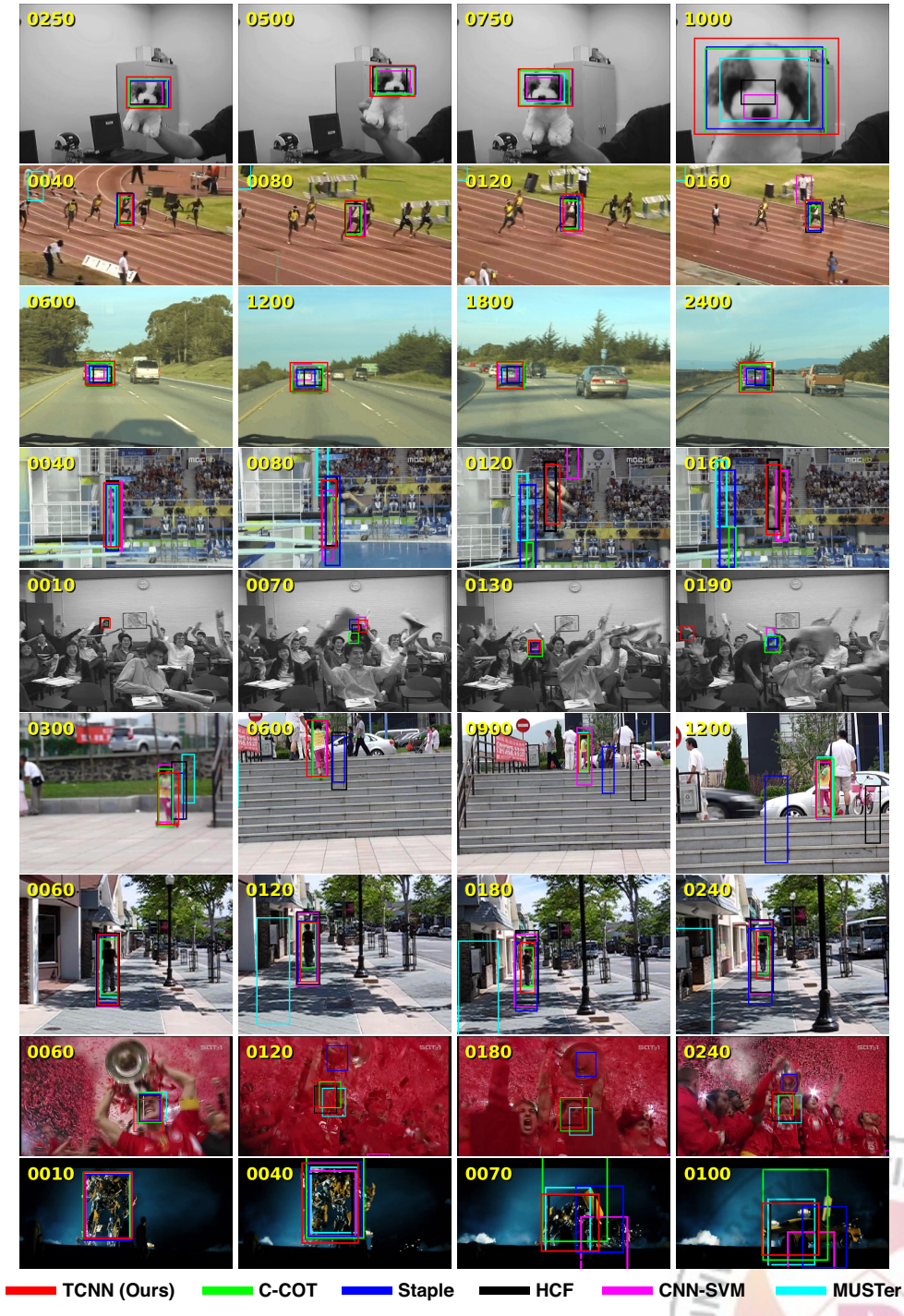


Figure 5.4: Qualitative results in several challenging sequences of OTB100 dataset (*Dog1*, *Bolt2*, *Car24*, *Diving*, *Freeman4*, *Girl2*, *Human9*, *Soccer*, and *Trans*).

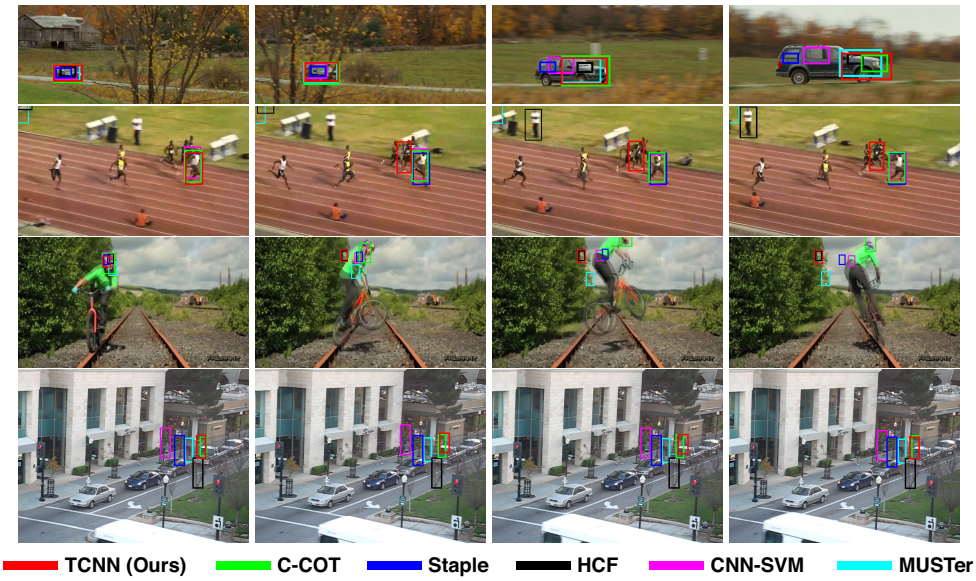


Figure 5.5: Failure cases of TCNN on OTB100 dataset (*CarScale*, *Bolt2*, *Biker*, *Human4*).

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

Table 5.1: Ablation studies. The rates of successfully tracked frames in terms of center location error with threshold 20 pixels and bounding box overlap ratio with threshold 0.5 are denoted by precision and success, respectively. AUC is based on the success plot in OTB evaluation protocol [1].

	Precision	Success	AUC
Single	85.6	79.9	63.1
Single-Stochastic	86.0	81.5	63.9
Linear	87.5	81.7	64.6
Tree	87.6	81.8	65.0
Tree-Stochastic	**88.2**	**83.0**	**65.3**

Table 5.2: AUC scores on OTB100 of TCNN with varying K and Δ.

K	1	5	10 (original)	20	50
AUC	0.639	0.652	0.653	0.653	0.653
Δ	1	5	10 (original)	20	50
AUC	0.598	0.630	0.653	0.629	0.624

5.2 Internal Analysis

The variations of proposed tracker are evaluated to analyze the effectiveness of each component. The following versions are implemented: a single CNN with sequential updates (Single), multiple CNNs with state estimation by simple averaging and sequential updates (Linear), and multiple CNNs with state estimation by weighted sum and tree updates (Tree). The postfix "Stochastic" denotes the stochastic model update scheme, while the others always make a new node for update. Table 5.1 summarizes the results in OTB100 from the internal comparison as follows. First, using multiple models (Linear) is more effective than using a single model (Single) because of the benefit from multi-modal appearances. Second, the approaches maintaining CNNs in a tree structure (Tree) enhances performance compared to the (Linear) since it allows each CNN to be updated more reliably through more appropriate paths. Finally, the proposed algorithm (Tree-Stochastic) is the most effective among all the other variations, which verifies that both model update and state estimation strategies are useful.

The impact of bounding box regression is also measured on the quality of the tracking algorithm. The representative accuracy in precision and success of TCNN without bounding box regression are (0.870, 0.634), while TCNN is (0.882, 0.653) in OTB100.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

Table 5.3: Expected average overlap score and the average scores and ranks of accuracy and robustness on the experiment in VOT2016 [2]. These are results on two separate settings for VOT2016, the *baseline* setting uses re-initialize protocol while the unsupervised setting does not. The first and second best scores are highlighted in red and blue colors, respectively.

Trackers	Baseline					Unsupervised		
	Accuracy		Robustness		Expected	Accuracy		Expected
	Rank	Score	Rank	Score	average overlap	Rank	Score	average overlap
Struck [9]	5.15	0.45	7.62	3.40	0.1416	6.62	0.24	0.3473
CCCT [42]	5.07	0.44	4.38	1.84	0.2230	5.10	0.31	0.4471
HMMTxD [43]	2.63	0.51	5.23	2.14	0.2313	4.07	0.37	0.4457
SiamFC-A [44]	3.17	0.53	4.53	1.91	0.2352	3.33	0.40	0.4818
SRDCF [39]	2.35	0.53	4.03	1.43	0.2471	3.25	0.40	0.4927
DeepSRDCF [26]	2.65	0.52	3.47	1.23	0.2763	3.02	0.43	0.4901
SiamFC-R [44]	1.58	0.55	4.03	1.36	0.2766	2.83	0.42	0.4963
EBT [45]	4.57	0.46	2.58	1.05	0.2913	4.23	0.37	0.4499
Staple [40]	2.10	0.54	3.97	1.42	0.2952	3.23	0.39	0.5088
C-COT [27]	2.18	0.54	2.27	0.89	0.3310	2.20	0.47	0.5276
TCNN	2.07	0.55	3.57	0.78	0.3419	2.43	0.47	0.5454

These result shows that bounding box regression is a useful component to alleviate the limitation of CNN-based features in object localization.

Table 5.2 shows the performance of TCNN with various parameters of maximum number of CNNs (K) and update intervals (Δ). The accuracy with respect to K is stable if $K \geq 5$, which is partly because the number of CNNs hardly exceeds 10 for most videos in testing dataset. It is because gradual update, which does not make a new CNN but finetunes existing one, tends to be selected 6 times often than drastic update. K is selected to restrict the maximum consumption of computational resources without losing performance. TCNN is not very sensitive to the variations of update interval Δ; its performance is highest at the current setting of $\Delta(= 10)$ and more frequent updates are not particularly helpful.

5.3 Evaluation on VOT2016 Dataset

I also test the proposed algorithm in the recent dataset for Visual Object Tracking challenge 2016 (VOT2016) [2], which is composed of 60 challenging video se-

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

quences with large variations. The VOT challenge provides re-initialization protocol, where trackers are reset with ground-truths in the middle of evaluation if tracking failures are observed. The performance metrics are defined based on accuracy and robustness, which are computed by the bounding box overlap ratio and the number of tracking failures. The VOT challenge also introduces the expected average overlap (EAO) as a new metric to rank tracking algorithms; it estimates how accurate the estimated bounding box is after a certain number of frames are processed since initialization.

TCNN is compared with 10 trackers, which include the half of the algorithms used for the evaluation in OTB since the results of the rest are unavailable in VOT2016 dataset. However, other good performing algorithms, *e.g.*, , EBT [45] and SiamFC [44] are included for comparison.

Table 5.3 illustrates the strength of TCNN in VOT2016 dataset compared to other methods. TCNN outperforms all the compared algorithms in the most representative metric, EAO[1], with significant margins. TCNN is also competitive consistently in all metrics, including results on *unsupervised* setting. Especially, TCNN has a substantial margin compared with other trackers with respect to EAO score, which is the primal metric of the VOT2016 dataset. Other CNN-based trackers, C-COT [27] and SiamFC-R [44] illustrate comparable results, but TCNN presents better accuracy with less tracking failures in general. EBT is robust since the tracker finds target in the entire image rather than a local search window, however, its overall score in terms of EAO is relatively low since its localization performance is not outstanding. I believe that TCNN achieves better performance by fully exploiting the representation power of deep features from multiple models.

[1]EAO score represents average overlap ratio of estimated target and ground-truth bounding boxes.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

Table 5.4: Expected average overlap score on the experiment for challenging attributes of VOT2016 [2]. The first and second best scores are highlighted in red and blue colors, respectively.

Trackers	Overall	Camera motion	Illum. change	Occlusion	Size change	Motion change	Not assigned
Struck [9]	0.1416	0.1491	0.1488	0.0957	0.1553	0.1107	0.0572
CCCT [42]	0.2230	0.2502	0.1721	0.1414	0.2182	0.1703	0.0638
HMMTxD [43]	0.2313	0.2887	0.1634	0.1912	0.1997	0.1721	0.0773
SiamFC-A [44]	0.2352	0.2357	0.1804	0.1615	0.2423	0.1907	0.0876
SRDCF [39]	0.2471	0.2350	0.1979	0.2177	0.2211	0.1695	0.1067
DeepSRDCF [26]	0.2763	0.3238	0.2938	0.1901	0.3144	0.2382	0.1062
SiamFC-R [44]	0.2766	0.2999	0.3757	0.1843	0.3013	0.2226	0.1235
EBT [45]	0.2913	0.3018	0.2247	0.2259	0.2640	0.2382	0.1092
Staple [40]	0.2952	0.3179	0.2691	0.2347	0.3122	0.2207	0.1362
C-COT [27]	0.3310	0.3539	0.4016	0.2462	0.3271	0.2490	0.1535
TCNN	0.3419	0.3322	0.3676	0.2994	0.3415	0.2903	0.1495

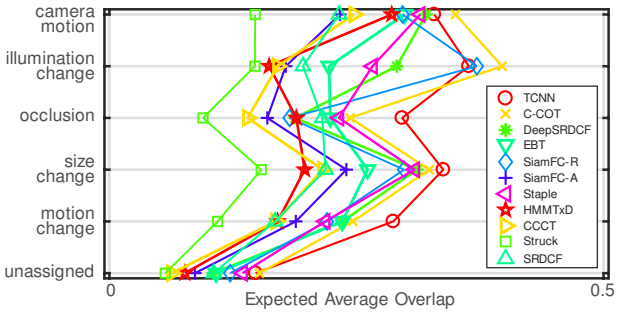
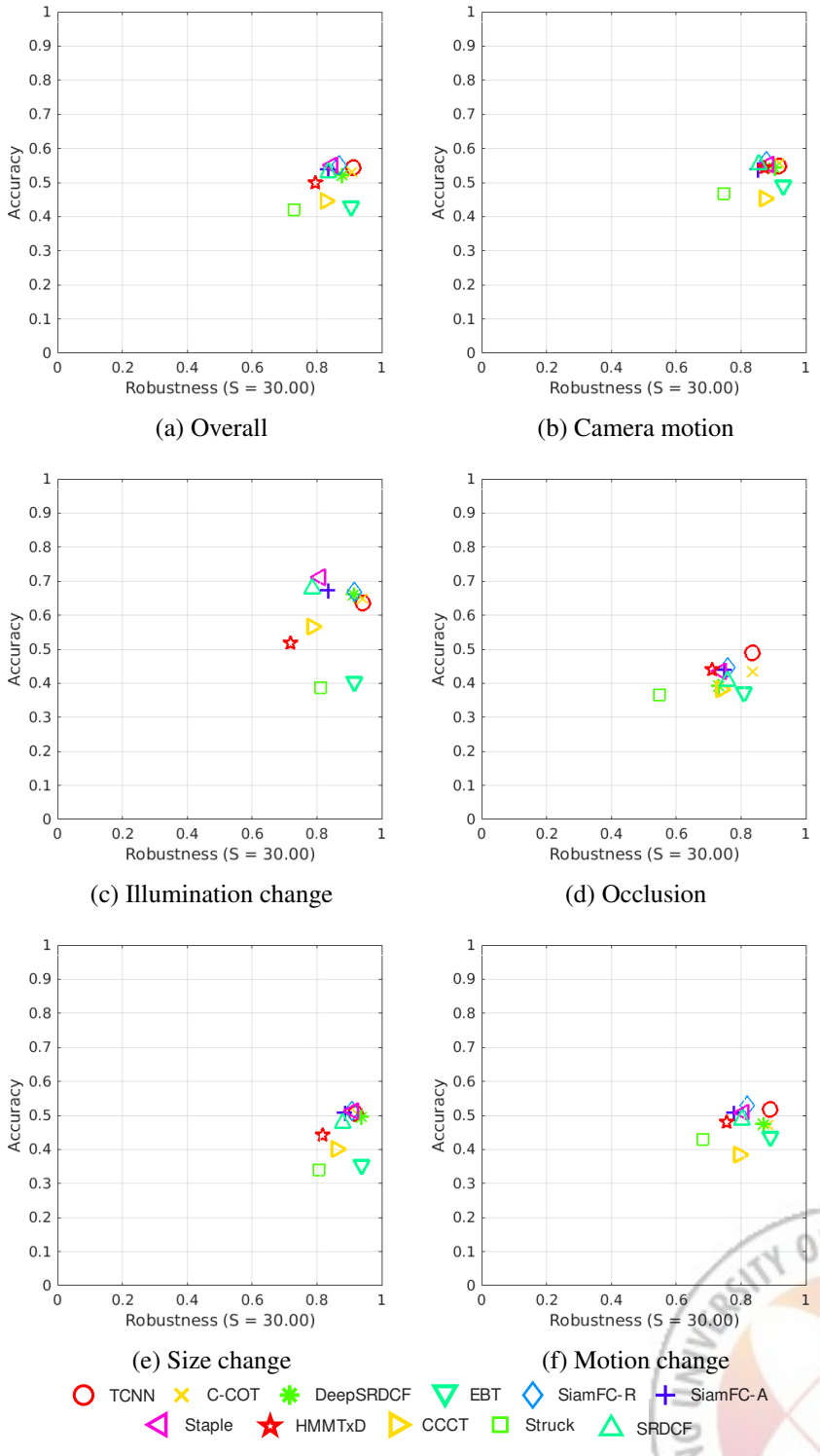


Figure 5.6: Quantitative results on VOT2016 [2] with respect to the challenging visual attributes. Trackers on the right side of the graph are better.

As same as the previous section, trackers are compared for various challenging attributes provided in the VOT benchmark. Results are shown on Table 5.4. TCNN shows good performance in most attributes while it is relatively weak for illumination changes; this is consistent with OTB result. Also, TCNN is especially good at difficult visual attributes such as occlusion and motion change, which shows the persistent and robust nature of the proposed method. Like the preceding evaluation results on OTB dataset, trackers with low-level features generally fail in more structural and semantic challenges such as motion change, camera motion and occlusion, which require high-level understanding about target and scene. You can clearly see the superior performance of TCNN over other algorithms on Figure 5.6, 5.7.



(a) Overall (b) Camera motion

(c) Illumination change (d) Occlusion

(e) Size change (f) Motion change

Figure 5.7: Quantitative results for 6 challenge attributes. Trackers on the upper-right side of the graph are better.

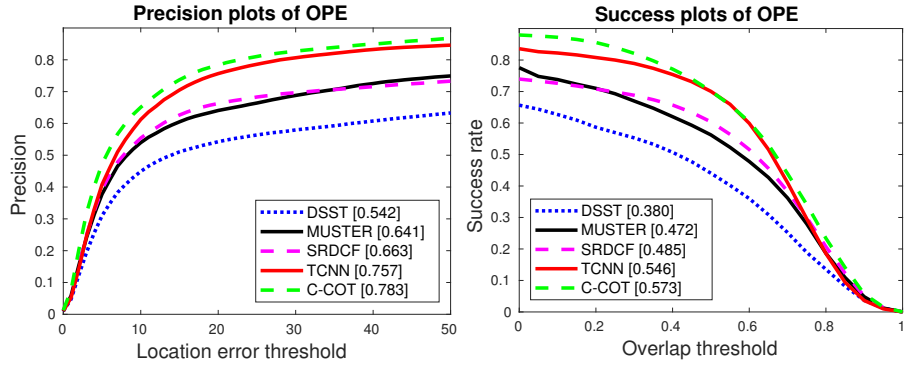


Figure 5.8: Quantitative results on TempleColor [3] dataset. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively.

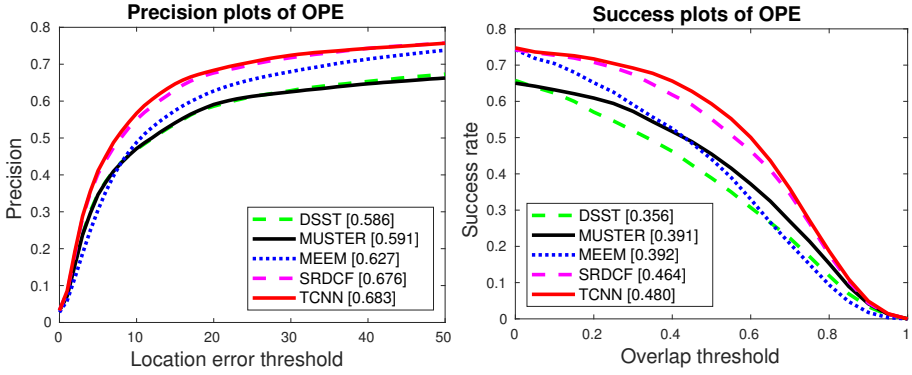


Figure 5.9: Quantitative results on UAV [3] dataset. The values in the legend are the precision at a threshold of 20 pixels and the AUC score, for precision and success plots, respectively.

5.4 Evaluation on TempleColor & UAV Datasets

For complete evaluation, TCNN is also compared with other state-of-the-art trackers on TempleColor [3] and UAV [38] datasets. TempleColor consists of 128 RGB video sequences where many sequences are overlapped with OTB dataset. UAV is a benchmark for low altitute UAV target tracking. As shown in the precision and success plots on Figure 5.8 and Figure 5.9, TCNN again shows comparable accuracy over other state-of-the-art trackers.

6 Conclusion

This work is a novel tracking algorithm based on multiple CNNs maintained in a tree structure, which are helpful to achieve multi-modality and reliability of target appearances. The state estimation of target is performed by computing a weighted average of scores from the multiple CNNs. The contribution of each CNN for target state estimation and online model update is also determined by exploiting the tree structure. Due to parameter sharing in convolutional layers, the use of multiple CNNs does not require substantial increase of memory and computation. Proposed tracking algorithm outperforms the state-of-the-art techniques in both OTB and VOT2016 benchmarks.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

요 약 문

본 연구는 복수의 외형 모델을 효율적으로 유지하는 컨볼루션 신경망 기반 물체 추적 알고리즘을 제안한다. 컨볼루션 신경망은 영상을 입력받아 컨볼루션 연산과 비선형 함수 입력을 여러 번 거듭하여 특징을 추출하는 알고리즘으로, 대량의 영상 분류 데이터를 이용하여 적합한 컨볼루션 연산의 인자를 찾는 기계학습 방식을 통하여 이미지 분류기의 정확도를 비약적으로 상승시켜 컴퓨터과학 전반에 큰 변혁을 일으키고 있다. 이렇듯 컨볼루션 신경망이 컴퓨터 비젼의 다양한 문제에 대한 좋은 해결책으로 부상하면서 이를 물체 추적 알고리즘에 적용하고자 하는 노력도 있어 왔다. 다만 대부분 이미 학습된 컨볼루션 신경망을 가져와 이를 특징 추출기로 이용하는 데 그친 바 있다. 이러한 한계를 넘어, 영상에서 물체를 추적함과 동시에 추적된 결과를 바탕으로 온라인 학습이 가능한 구조를 고안해 보았다. 컨볼루션 신경망을 여러 개 유지하되, 시간 순서에 맞추어 선형적으로 추가하는 대신 트리 구조를 이용하여 영상 내의 각기 다른 구간의 영상을 기반으로 추계적으로 학습하는 것이 그 골자이다. 여러개의 컨볼루션 신경망을 여러 개의 브랜치를 가진 트리 구조 내에 저장함으로써 추적 대상의 다양한 외형 모델을 잘 유지하면서도 안정적인 업데이트가 가능하다. 이러한 구조는 일부 신경망이 외형 모델을 잘 담지 못하게 되더라도, 다수의 다른 신경망의 정확한 판단으로 인해 끈질기고 안정적으로 물체를 추적할 수 있는 장점이 있다. 각 컨볼루션 신경망은 대다수의 인자를 서로 공유함으로서 그 갯수가 늘어나더라도 계산양과 메모리 요구량이 선형적으로 증가하지 않도록 설계했다. 물체 추적기의 정확도를 측정하는데 두루 사용되는 다양한 물체 추적 벤치마크 영상군에서 (OTB, VOT) 실험해 본 결과, 본 알고리즘이 좋은 성능을 보이는 것을 확인할 수 있었다.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교

Bibliography

[1] Y. Wu, J. Lim, and M. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[2] M. Kristan and et al., "The visual object tracking VOT2016 challenge results," in *ECCVW*, 2016.

[3] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, pp. 5630–5644, Dec 2015.

[4] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *ECCV*, 2014.

[5] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *ICCV*, 2009.

[6] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*, 2012.

[7] B. Han and L. S. Davis, "On-line density-based appearance modeling for object tracking," in *ICCV*, 2005.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

[8] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *BMVC*, 2006.

[9] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2011.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[13] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015.

[14] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *The psychology of learning and motivation*, vol. 24, no. 109-165, p. 92, 1989.

[15] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations by convolutional neural networks for visual tracking," in *BMVC*, 2014.

[16] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[17] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

[18] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, pp. 2544–2550, June 2010.

[19] 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.

[20] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.

[21] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking," in *CVPR*, 2015.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database a large-scale hierarchical image database," in *CVPR*, 2009.

[23] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2015.

[24] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015.

[25] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *ICML*, 2015.

[26] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *ICCVW*, 2015.

[27] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *ECCV*, 2016.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

[28] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *ECCV*, 2016.

[29] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR*, 2010.

[30] H. Nam, S. Hong, and B. Han, "Online graph-based tracking," in *ECCV*, 2014.

[31] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *ICCV*, 2007.

[32] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[33] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, "Randomized ensemble tracking," in *ICCV*, 2013.

[34] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.

[35] S. Hong and B. Han, "Visual tracking by sampling tree-structured graphical models," in *ECCV*, 2014.

[36] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *BMVC*, 2014.

[37] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *ACM MM*, 2015.

[38] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *ECCV*, 2016.

[39] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1986 포항공과대학교

[40] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[42] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *CVPR*, 2013.

[43] T. Vojir, J. Matas, and J. Noskova, "Online adaptive hidden markov model for multi-tracker fusion," *Computer Vision and Image Understanding*, vol. 153, pp. 109 – 119, 2016. Special issue on Visual Tracking.

[44] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCVW*, 2016.

[45] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specic proposals," in *CVPR*, 2016.

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
1986
포항공과대학교