Globally Optimal Object Tracking with Complementary Use of Single Shot Multibox Detector and Fully Convolutional Network

Jinho Lee

2018.2

Five-year Doctoral Course

Department of Systems Life Sciences

Bioinformatics Course

Contents

1 Introduction **1**

2 Related Work **4**

3 The Proposed Method **6**
3.1 Likelihood Map by Several Neural Networks 6
3.1.1 Likelihood Map by Convolutional Neural Network 7
3.1.2 Likelihood Map by Fully Convolutional Network 9
3.1.3 Likelihood Map by Single Shot Multibox Detector 11
3.1.4 Likelihood Maps by Combining Single Shot Multibox Detector and Fully Convolutional Network 13
3.2 Global Path Optimization by Dynamic Programming 14
3.3 Synergies by Combining SSD, FCN, and DP 16

4 Implementation and Experiments **17**
4.1 Experimental Setup 17
4.1.1 Dataset 17
4.1.2 Experimental Sequences 17
4.2 Evaluation 18
4.2.1 Evaluation Criterion 18
4.2.2 Comparison Trackers 19
4.3 Experimental Results 19
4.3.1 Result without Initialization 19
4.3.2 Result with Initialization 20
4.3.3 Result Depending on Various Methods of Obtaining likelihood Map 22
4.3.4 Various Result Examples 23

4.3.5 Superiority over Fully Convolutional Network Based Tracker 24

5 Conclusion **29**

Acknowledge **33**

Bibliography **34**

Chapter 1

Introduction

Object tracking is defined as problem of estimating spatio-temporal trajectory of a target object in an image. Although it has been studied for many applications such as bio-image analysis, scene surveillance, autonomous vehicle control, etc, it is still a difficult problem. One difficulty comes from appearance variation. As shown in Fig. 1.1, for a general person tracking problem, we need to deal with various clothes, poses, and body shapes under various illumination condition. Traditional methods assume a predefined template of the target object and update it accordingly to any changes in appearance [1, 2]. Another difficulty is occlusion. Traditional object tracking methods are often intolerant to severe occlusion [3, 4, 5].

In this paper, we propose a object tracking method which is robust to both appearance variation and occlusion by using a complementary combination of Single Shot Multibox Detector (SSD) [6], Fully Convolutional Network (FCN) [7], and Dynamic Programming (DP) [8]. SSD and FCN are employed for tackling appearance variation. They have been proposed recently for object detection and segmentation. They can also provide a probability value of a target object for each category, such as person, car, and motorbike,

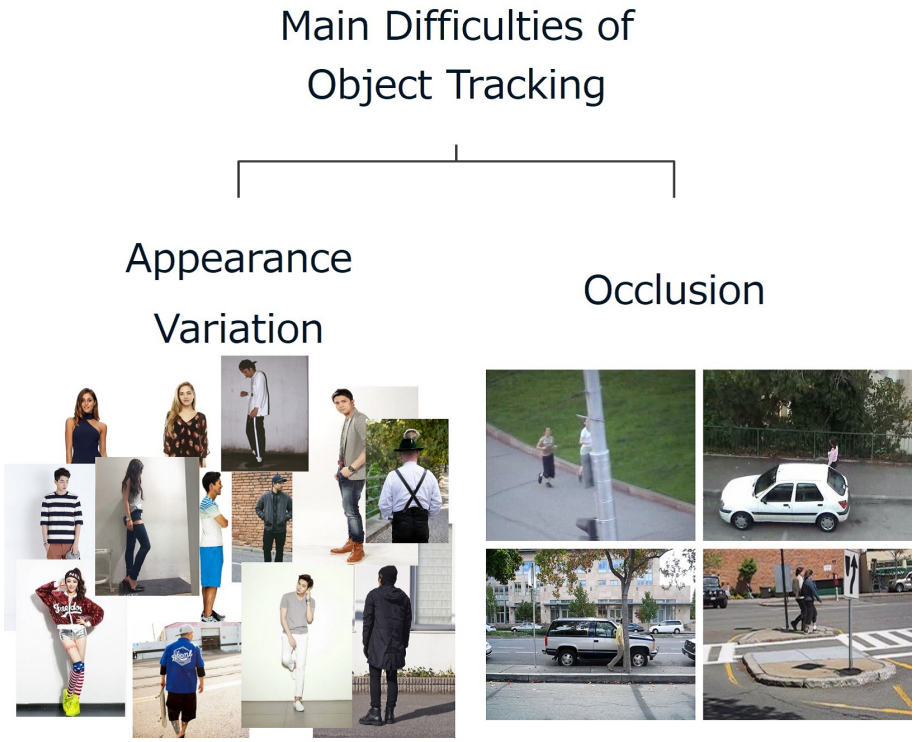

Figure 1.1 Main Difficulties of Object Tracking are appearance variation and occlusion. Traditional trackers without parameter changes are intolerant to track target object with various clothes, poses, and body shapes and also difficult to deal with severe occlusion.

given each bounding box or pixel respectively. Since SSD and FCN are types of CNNs, large amounts of training samples will make them robust to variety of appearances.

To deal with occlusion, we utilized DP for global optimization of a target object's trajectory. DP is one of the most fundamental optimization techniques and has been used for obtaining a globally optimal tracking path. Since a slope constraint of DP prohibits the tracked position from moving steeply over all frames, it is possible to obtain a stable tracking path, regardless of occlusion.

It is very important to note the reason why we use the two CNN-based object detectors,

SSD and FCN, in a complementary manner, is because they provide detection results in different ways. SSD provides an accurate detection result for a clear target object, however it is also impossible to provide a detection result in an unstable situation such as occlusion. In contrast to SSD, FCN provides a result, regardless of any situation. Namely, combination of SSD and DP is useful to stable situation to obtain accurate result and it of FCN and DP is utilized in unstable situation to obtain any result.

It is also noteworthy that the proposed method requires neither the initial position nor the template of the target object. Traditional trackers may be sensitive to the template of the target object and the initialization in which the initial position of the target object is denoted on the first frame. However, the proposed method does not require either the template nor the initialization because of the synergy combining SSD, FCN, and DP.

The contributions of this paper are as follows. First, we show the proposed method achieved the highest accuracy compared to the traditional trackers introduced in the Visual Tracker Benchmark [11]. Second, we confirm that the complementary use of the two CNN-based object detectors, SSD and FCN, are useful for tracking. Third, we confirm that the proposed method tackles appearance variation and occlusion through several experiments even without initialization, templates, and modifying parameters.

The remaining of this paper is organized as follows. In Section 2, we introduce related traditional tracking research. Section 3 elaborates on SSD, FCN, and DP and details the proposed method. In Section 4, we confirm that the proposed method is a robust tracker through several experiments and analyze the experimental results. Finally, Section 5 draws the conclusion.

Chapter 2

Related Work

Object tracking is one of the important techniques in computer vision and has been actively studied for decades. Most object tracking algorithms are divided into two categories: generative and discriminative methods. Generative methods describe appearance of a target object with using a generative model and search for the target object region that fits the model best. A number of generative model based algorithms have been proposed such as sparse representation [12, 13], density estimation [14, 15], and incremental subspace learning [16]. On contrary, discriminative methods build a model to distinguish a target object from the background. These tracking methods include P-N learning [17] and online boosting [18, 19, 20]. Even though these approaches are satisfactory in restricted situations, they have inherent limitations which include occlusion and appearance variation such as illumination changes, deformation etc.

To deal with limitations which traditional trackers can not tackle, recent trackers employ Convolutional Neural Networks (CNN) [21, 22] and Deep Convolutional Neural Networks (DCNN) [23, 24] by focusing on their powerful performance. A number of trackers using neural networks have been proposed such as human tracking, hand tracking,

Table 2.1: Comparison between the proposed method and FCNT [29].

Tracker	Target	Template	Initialization	Path optimization	Offline/Online
Proposed	General	Unnecessary	Unnecessary	Globally Optimal by DP	Offline
FCNT	Specific	Necessary	Necessary	Greedy	Online

etc. [25, 26, 27, 28]. Representative tracker using a neural network is a Fully Convolutional Network based Tracker (FCNT) [29] which also utilizes FCN. This method utilizes multi-level feature maps of a VGG network [30] to complement drastic appearance variation and distinguish a target object from its similar distracters. It selects discriminative feature maps and discards noisy ones, because the CNN features pretrained on ImageNet [23] are for distinguishing generic objects. Even though FCNT achieved a high accuracy compared to conventional trackers, initialization and templates are necessary to track a target object.

Table 1 shows the comparison of characteristics between the proposed method and FCNT. The main difference between the proposed method and FCNT is whether initialization and templates of a target object are necessary or not. Namely, FCNT can track only a specific target object with defined initial position and template. In contrast, it is possible to use the proposed method without them.. The other difference is that FCNT uses a greedy tracking algorithm whereas the proposed method utilizes DP for globally optimal tracking. In Section 4.3.5, we will prove experimentally that the proposed method has superiority over FCNT.

Chapter 3

The Proposed Method

The proposed method is largely divided into two steps. First step is to generate likelihood map by SSD and FCN. Second one is to obtain globally optimal tracking path by DP. In this chapter, we elaborate on details of each step and also draw synergies by combining SSD, FCN, and DP.

3.1 Likelihood Map by Several Neural Networks

Likelihood map is defined as two-dimensional probability distribution of a target object position at a certain frame. A peak in a likelihood map at frame t suggests a candidate position of the target object at t. In the proposed method, we utilize both SSD and FCN to obtain likelihood map, however, the other neural networks are also possible to obtain likelihood map. To explain why combining of SSD and FCN is more suitable to obtain likelihood map than the others, we elaborate on process of obtaining likelihood map of each neural network (Convolutional Neural Network, Fully Convolutional Neural Network, Single Shot Multibox Detector) and details of combining of SSD and FCN as well.

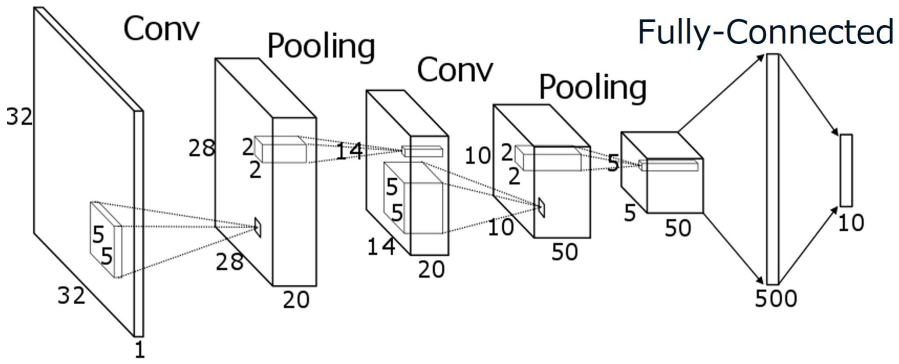

Figure 3.1 Basic Structure of CNN: CNN is largely comprised of convolutional layer, pooling layer, and fully-connected layer. Through passing these layers, CNN can extract various features.

3.1.1 Likelihood Map by Convolutional Neural Network

Convolutional Neural Network (CNN) is a powerful neural network which is composed of convolution, pooling layers. It has shown significant performance in image processing field. Combined with recent method such as dropout layer and ReLU (Rectified Linear Units), CNN models have outperformed most of traditional methods.

Fig. 3.1 shows the basic structure of CNN. CNN receives an input and process it by a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer. Commonly, CNN involves three main layers: convolutional layer, pooling layer, and fully-connected layer. The convolutional layer apply convolution operation to input and pass the result to the next layer. Namely, it calculates a dot product between weights and connected small regions and pass results to next layer. Pooling layer reduces the spatial size of the representation to reduce the amount of parameter and computation. Through adding fully-connected layer, CNN is possible to learn non-linear combination of features by various activation function, although the convolutional and pooling layer provide somewhat invariant features.

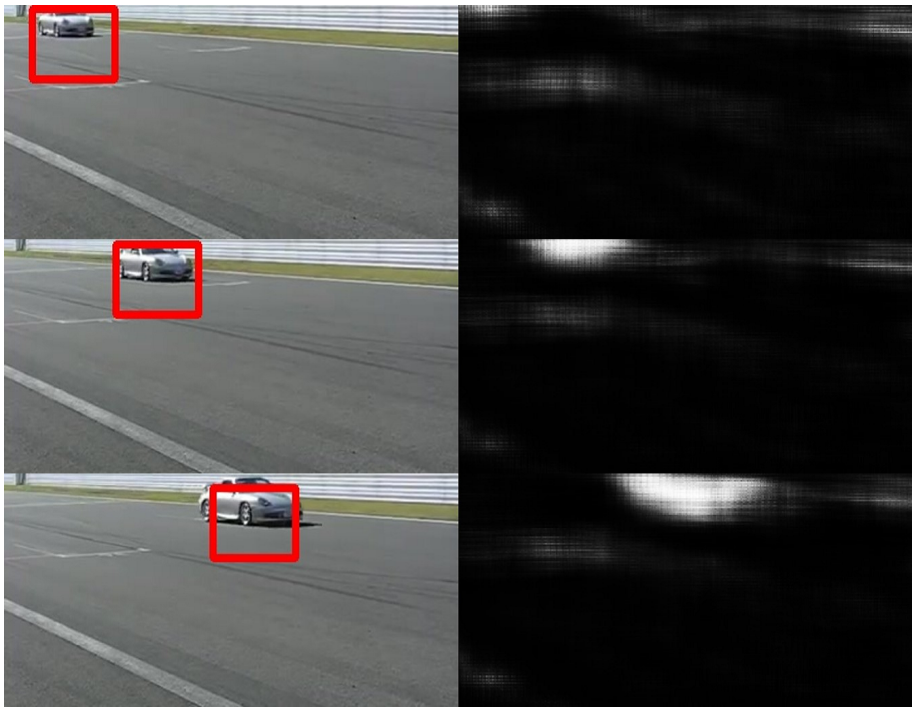

Figure 3.2 Example results by single CNN.

To obtain likelihood map by single CNN, we input a fixed-size block image to CNN by sliding over a frame. The CNN is made of 3 convolutional layers of 32, 32, and 64 nodes respectively with 5×5 kernels and Cifar10 dataset is utilized for training the CNN, which consists of 6000 32×32 color images for each of 10 object classes. We utilize the Softmax for output layer. The probability value of each pixel is obtained by setting the output of CNN to center position of the block image.

Fig. 3.2 shows example results with input frames (440×220) and correspondingly-sized likelihood maps. The white color of the likelihood maps means high value, in contrast with the black color. The red square indicates the sliding fixed block image (80×60).

This method is very naive and easy to implement, however there are largely two defects by single CNN. First is time cost problem. Since likelihood map by single CNN is obtained by inputting block image to CNN with sliding over the frame, it requires many

forward calculations. To obtain 440×220 likelihood map with 80×60 sliding block image, it requires 40 sec. per frame. Another defect is weakness of size change of a target object. Since this method accepts a fixed size region as input, it can not deal with a target object of various size.

3.1.2 Likelihood Map by Fully Convolutional Network

Fully convolutional network is powerful visual model that output hierarchies of features. FCN is composed entirely of convolutional layers based on VGG-16 network, as shown in Fig. 3.3. FCN is trained by end-to-end, pixels-to-pixels and outperformed the state of the art technologies in segmentation field. It can accept input of arbitrary size and produce a correspondingly-sized likelihood map by up and down sampled pooling layers.

The likelihood map by FCN might include noisy probability values by up and down sampled pooling layers. To obtain accurate positive response of a target object, links between the low-level fine layers and the high-level coarse layers are constructed. These are so called *skip connection* which combines information from fine layers and coarse layers, as shown in Fig. 3.4.

Fig. 3.5 shows examples of likelihood map by FCN. As shown in Fig. 3.5, even if we can obtain a more accurate likelihood map by Skip Connection, the likelihood map obtained by FCN still includes noisy probability values.

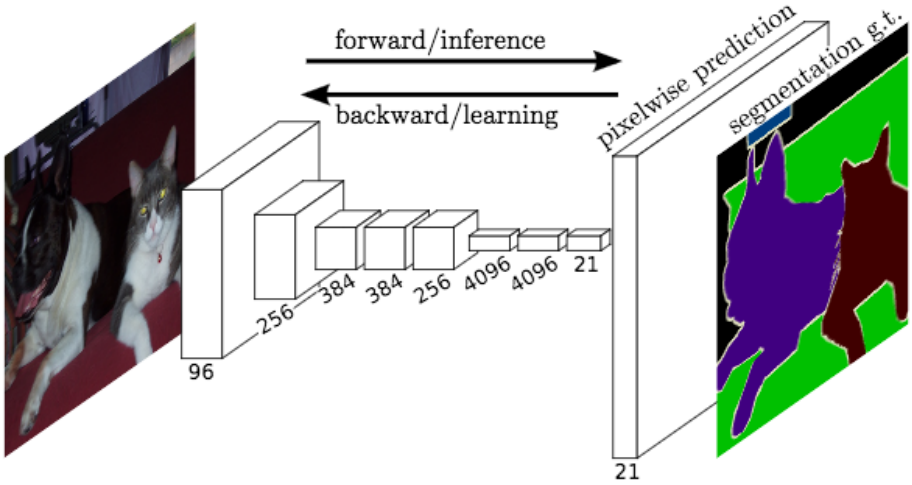

Figure 3.3 FCN Network Model.

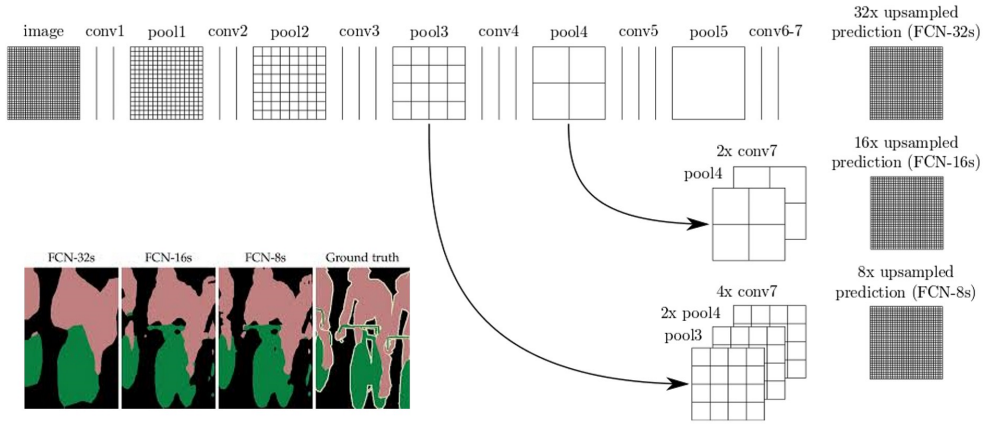

Figure 3.4 Skip Connection: links between the low-level fine layers and the high-level coarse layers can generate the more high-resolution likelihood map.

Figure 3.5 Examples of Likelihood Map by FCN: Left side of couple image is input image, and right one is likelihood map by FCN. FCN can extract the pixel-wise features, however, likelihood map by FCN contains lots of noise, despite of Skip Connection.

3.1.3 Likelihood Map by Single Shot Multibox Detector

Single Shot Multibox Detector (SSD) is a method to detect objects in images by using a single neural network and based on VGG-16 network which includes 13 convolution layers and 3 pooling layers, as shown in Fig. 3.6. It possesses supplementary two characteristics: convolutional predictors and multi-scale feature maps. The convolutional predictors generate a probability value for the presence of each object category in each default box and produce adjustments to the box to match the object shape. Additionally, the network combines predictions from multi-scale feature maps with different resolutions to handle objects of various sizes.

We generate a likelihood map by setting a probability value, on the center position of resulting bounding box of SSD. Thus, likelihood maps obtained by SSD contain a very accurate probability value. However, when SSD fails to detect a target object, likelihood maps can not be obtained. Fig. 3.7 shows examples of SSD result and likelihood map by SSD.

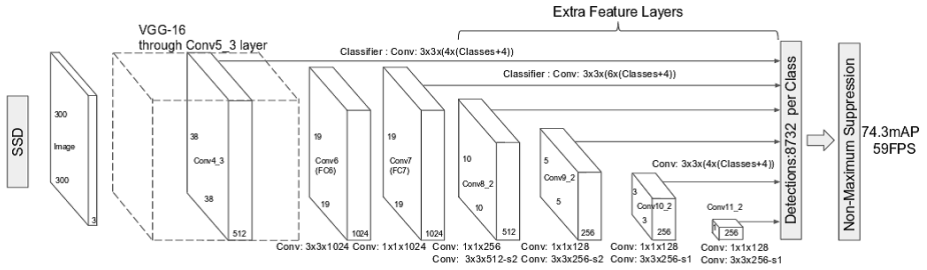

Figure 3.6 SSD Network Model.

Figure 3.7 Examples of SSD result and likelihood map by setting a probability value to the center position of resulting bounding box: Examples of two rows from top are success example. And it of last row is failure example when the target object is occluded by obstacle.

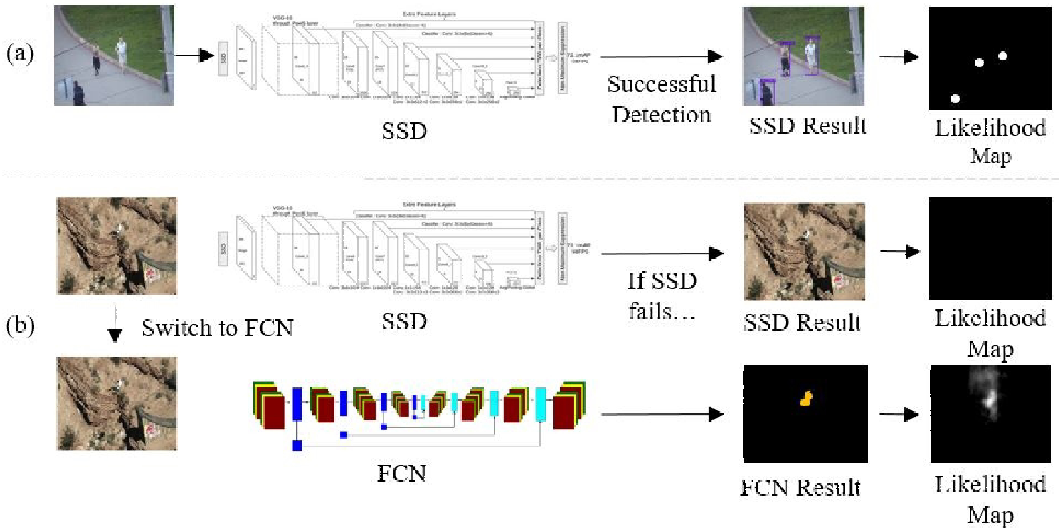

Figure 3.8 Pipeline to generate a likelihood map from an input image: (a) shows the process to generate a likelihood map by SSD. The likelihood map by SSD provides accurate probability values and positions of target objects when the targets are rather easy for detection. When SSD fails to detect the target, we switch to FCN and employ the likelihood map by FCN, as shown in (b). The likelihood map by FCN might include noisy probability values, compared to it by SSD.

3.1.4 Likelihood Maps by Combining Single Shot Multibox Detector and Fully Convolutional Network

In the proposed method, both SSD and FCN are used for obtaining likelihood maps. Fig. 3.8 shows the pipeline of how to obtain a likelihood map from an input image. We will switch two neural networks according to the situation as shown in Fig. 3.8. This is because SSD and FCN shows different behaviors especially when object candidate detection is difficult, as follows.

Using both SSD and FCN to obtain a likelihood map increases the tracking accuracy.

Although SSD is a detection method with high accuracy, it might not detect the target object in unstable situations such as occlusion, blurriness, and deformation, as shown in (b) of Fig. 3.8. If SSD fails to detect, we switch to FCN and obtain likelihood maps by FCN. The success and failure criteria of detection by SSD is whether the detected position exists within N pixels from the highest value position of the previous frame or not. FCN provides likelihood maps for all input images, regardless of unstable situation, even if they might include noisy probability values. Note that, as discussed later, even when both SSD and FCN cannot obtain likelihood values (e.g. when an target object that has been tracked leaves the scene), DP complements the tracking path.

The other merit of using SSD and FCN is their computational efficiency. A naive method to obtain likelihood maps is to apply a CNN to a sliding window region of an input image, as discussed in Section 3.1.1. Using this method requires many forward calculations and can not deal with a target object of various sizes, because it accepts a fixed region size as input. However, both SSD and FCN accept the entire image and only require a single forward calculation with handling various sizes.

3.2 Global Path Optimization by Dynamic Programming

To apply Dynamic Programming (DP) to our method, we start by creating likelihood maps of each of the frames using SSD or FCN. Fig. 3.9 shows the procedure to obtain the most optimal tracking path by DP. For each pixel on a likelihood map, we find the highest value within a given slope constraint of the previous frame which prohibits from moving steeply and create cumulative DP maps. This process is continued by iterating

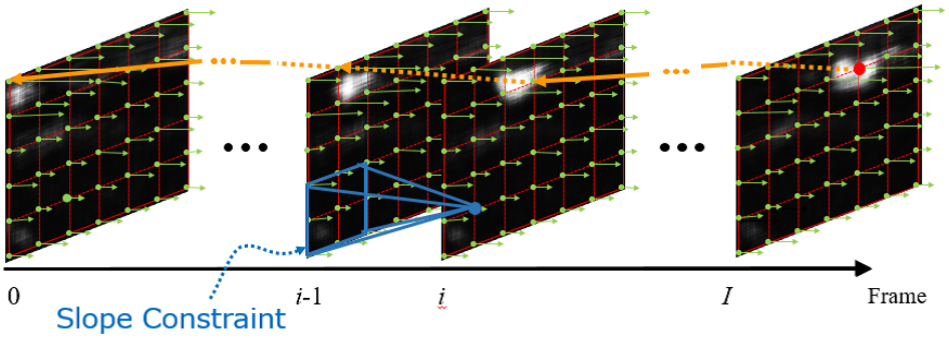

Figure 3.9 The tracking path optimization in the proposed method: Each frame is a likelihood map calculated by SSD or FCN and the green arrow suggests a probability value of a target object at each position. The blue rectangle is a slope constraint to restrict movement and the red plot is the position where the sum of probability values over the associated path is highest at the final frame. The orange arrows mean the most globally optimal tracking path obtained by back-tracking.

over all of the frames. Cumulative DP map $D^{(f)}$ is defined as:

$$D^{(f)}(x,y) = \max_{x-w_s \le x \le x+w_s, y-h_s \le y \le y+h_s} [D^{(f-1)}(x,y)] + L^{(f)}(x,y) \tag{3.1}$$

where likelihood map is $L^{(f)}$, f is the number of frame and size of slope constraint is denoted as (w_s, h_s). We select the highest probability value on the final cumulative DP map. After that, DP searches for the most optimal tracking path by back-tracking along the highest probability value on each previous likelihood map. DP is a non-greedy algorithm to estimate the global optimal path in a sequence. Due to this, DP-based tracking is robust to occlusion, which degrades a tracking performance of greedy algorithms.

3.3 Synergies by Combining SSD, FCN, and DP

We propose combination of SSD, FCN, and DP as a robust object tracking method. The proposed method has not only advantage of robustness to appearance variation and occlusion, but also does not need to set a template, change initialization, or change parameters, even when appearance of a target object changes.

The proposed method does not need a template for object tracking. For traditional trackers, a template is necessary and needs to be updated when a target object changes. However, for the proposed method, it is unnecessary, under the condition that a category of a target object is trained sufficiently. Since, the proposed method can deal with appearance variation by learning numerous features of a target object, it also does not need to modify parameters even if appearance of a target object is changed. For traditional trackers, identifying position of a target object on the first frame is important element to track. However, the proposed method can obtain the most globally optimal tracking path by back-tracking over all cumulative DP maps without any identifying the position.

Chapter 4

Implementation and Experiments

4.1 Experimental Setup

4.1.1 Dataset

We used the VOC2012 [31] dataset to train SSD and FCN on 20 categories. The training dataset has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations. The categories are as follows: *person, bird, cat, cow, dog, horse, sheep, airplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa and tv/monitor*, as shown Fig. 4.1.

4.1.2 Experimental Sequences

To demonstrate that the proposed method can track a target object with a high accuracy, we evaluated the proposed method using sequences that have a target object in one of the 20 categories of VOC2012. Since the proposed method can detect only trained objects on the 20 categories, we selected 12 sequences for our experiments: *CarScale, Coke, Cou-*

Figure 4.1 20 classes examples of VOC2012 Dataset.

ple, Crossing, David3, Jogging1&2, MotorRolling, MountainBike, Walking1&2, Woman. It is noteworthy that those sequences show various difficulties, such as illumination variation, scale variation, occlusion, fast motion, rotation, and low resolution.

The sequences were classified into two types, single-object sequences[1] and multi-object sequences[2] . Since the proposed method is designed to track a single object without initialization, single-object sequences are appropriate for performance evaluation. The proposed method, however, is still applicable to multi-object sequences by initialization. We therefore conducted two separated experiments, single-object sequences (without initialization) and multi-object sequences (with initialization).

4.2 Evaluation

4.2.1 Evaluation Criterion

We evaluated the proposed method by comparing the precision which is established method in the Visual Tracker Benchmark [11]. The precision is defined as the percentage of frames whose estimated position is within a given threshold from a ground-truth. The distance between the estimated position and the manually labeled ground-truth is

[1]single-object sequences contain *CarScale, Coke, Couple, Crossing, David3, MotorRolling, MountainBike, Walking1&2, Woman.*

[2]multi-object sequences contain *CarScale, Coke, Couple, Crossing, David3, Jogging1&2, MotorRolling, MountainBike, Walking1&2, Woman.*

calculated by Euclidean distance. To show a performance efficiently, we conducted one-pass evaluation (OPE) that trackers run throughout a test sequence only one time and compare the precision of each of trackers.

4.2.2 Comparison Trackers

Tracking methods can be divided into offine tracking such as the proposed method and online tracking. However, we compared the proposed method to online tracking methods in order to show the performance, because there is no comparable offline tracking methods which are released. We compared the proposed method to the top five traditional trackers introduced in the Visual Tracker Benchmark: Structured Output Tracking with Kernels (Struck) [32], a sparsity-based tracker (SCM) [33], P-N Learning tracker (TLD) [34], Context tracker (CXT) [35] and Visual Tracking Decomposition (VTD) [36].

4.3 Experimental Results

4.3.1 Result without Initialization

As i mentioned in Section 3.3, the proposed method does not require initialization which is identifying the initial position on the first frame through combining SSD, FCN, and DP. Fig. 4.2 shows all precision results including the proposed method without initialization and the traditional trackers with initialization. The score listed in the legend of Fig. 4.2 is the precision at a threshold of 20 pixels, since a 20 pixel threshold is the standard threshold for the Visual Tracker Benchmark. As shown in Fig. 4.2, we confirmed that the proposed method outperforms than the traditional trackers, even though the proposed method is not given the initial position of the target object on the first frame.

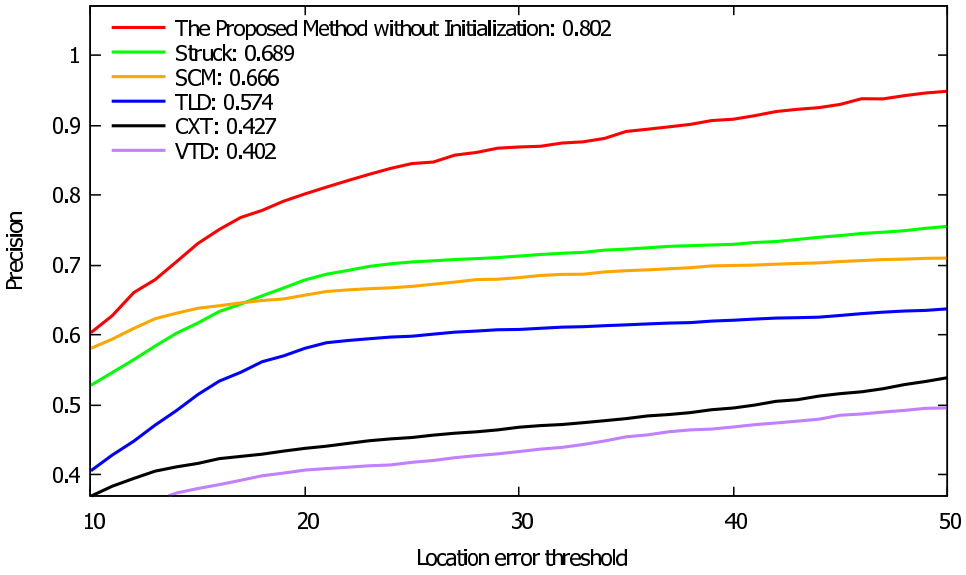

Figure 4.2 Precision results for single-object sequences: We compared results of the proposed method without initialization and those of the traditional trackers with that by using single-object sequences. The precision is defined as the percentage of frames whose estimated position is within location error threshold. The permissible threshold distance is denoted as location error threshold. The score listed in the legend means the precision score at a threshold of 20 pixels. The proposed method possesses the higher performance for all thresholds and has no large deviation from ground-truth.

Since DP sets a slope constraint to prohibit tracked position from moving rapidly, the proposed method can track a target object with small deviation.

4.3.2 Result with Initialization

As we mentioned in Section 4.1.2, the proposed method is applicable to multi-object sequences by identifying a initial position of a target object on the first frame. Fig. 4.3 shows all results with initialization. For all thresholds, the proposed method possesses

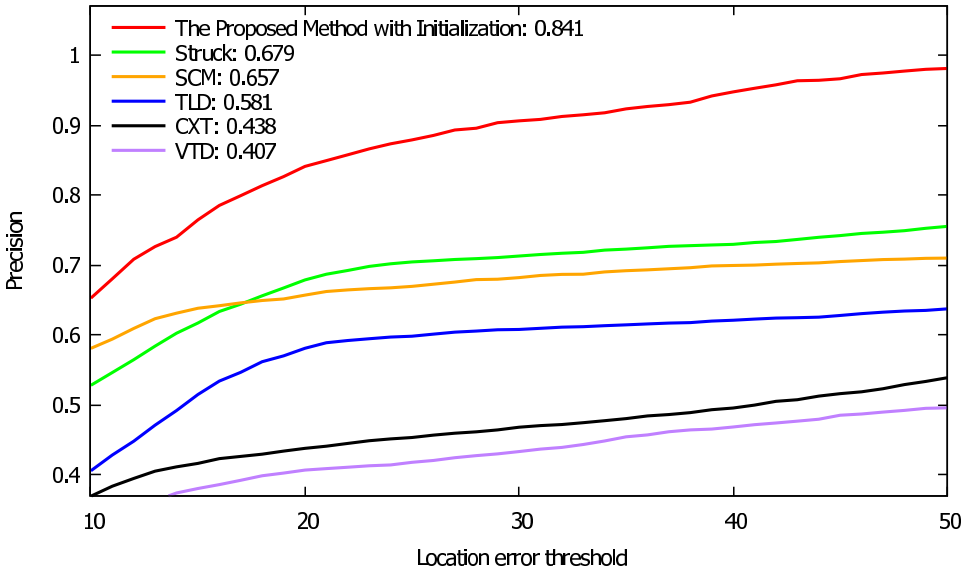

Figure 4.3 Precision results for multi-object sequences: We summarized the all results with initialization by using multi-object sequences. The precision is defined as the percentage of frames whose estimated position is within location error threshold. The permissible threshold distance is denoted as location error threshold. The score listed in the legend means the precision score at a threshold of 20 pixels. The proposed method possesses the higher performance for all thresholds with initialization as well.

a higher performance compared to the traditional trackers. Through these results, when multi-objects of same class even exist on the same frame, the proposed method can track a target object distinguishably by initialization. By comparing the results of Fig. 4.2 and Fig. 4.3, we also can confirm that the precision of the proposed method with initialization is more accurate than that without initialization entirely.

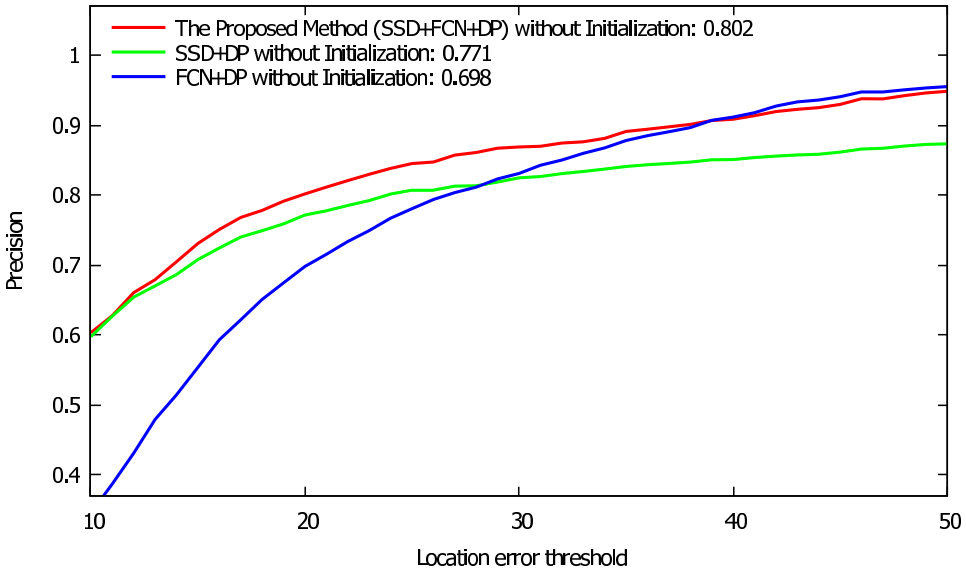

Figure 4.4 Precision results according to how to generate a likelihood map: We compared the results using likelihood maps by both SSD and FCN, single SSD, and single FCN, without initialization. The precision is defined as the percentage of frames whose estimated position is within location error threshold. The permissible threshold distance is denoted as location error threshold. The score listed in the legend means the precision score at a threshold of 20 pixels. The proposed method which is using likelihood maps by both SSD and FCN, possesses a higher performance than the others.

4.3.3 Result Depending on Various Methods of Obtaining likelihood Map

We also conducted additional experiment to compare the results using a likelihood map by both SSD and FCN, single SSD and single FCN. Fig. 4.4 shows the precision results using likelihood maps by both SSD and FCN, a single SSD, and a single FCN, respectively, without initialization. The proposed method which utilizes both SSD and FCN has a higher performance than the others at a threshold of 20 pixels. Since the

method using single SSD can not generate likelihood maps for all input images, the results by single SSD are worse than those by the proposed method. As shown in Fig. 4.4, the precision by single FCN ascends rapidly for low thresholds. When a target object is large enough, the method using single FCN might track the position which is far from the center position of a target object, because FCN does not always obtain the highest probability value which is close to center position of a target object. Due to this, the precision by single FCN is lower than the others for low thresholds.

4.3.4 Various Result Examples

Fig. 4.5, 4.6, 4.7 show examples of the proposed method dealing with appearance variation and occlusion. As shown in these results, although there are various target objects of same category in each sequence, the proposed method can track each target object without template and parameter modification. Also, we confirmed that the proposed method can track a occluded target object more stably because DP seeks the global optimal path over all frames. However, when target objects of same category appear with occlusion such as *Walking 2* sequence in Fig. 4.7, the proposed method confused which object should track. Since the proposed method utilizes somewhat general features of same category during tracking, we should utilize specific features of a target object to track distinguishably. Also, we confirmed that the proposed method is applicable with initialization, through *Jogging, Jogging2* sequences.

4.3.5 Superiority over Fully Convolutional Network Based Tracker

The last experiment is the comparison experiment between the proposed method and Fully Convolutional Network Based Tracker (FCNT) [29] which is a tracker using FCN, to demonstrate superiority of the proposed method than FCNT. We also observed the performance of FCNT [29] using the same sequences. It could achieve 0.951 and 0.945 precision for single-object and multi-object sequences, respectively, at threshold of 20 pixels. It is, however, almost meaningless to compare this precision to ours. First of all, we should remember that FCNT needs a template and ours does not. In addition, FCNT needs initialization and ours does not. In fact, we can see that the proposed method without initialization has no severe degradation from comparison between Fig. 4.2 and 4.3. Furthermore, our DP-based method has theoretical superiority over FCNT at the robustness to occlusion, as shown in Fig. 4.8.

Figure 4.5 Example results of the proposed method dealing with appearance variation and occlusion (1): The center of red '+' means the tracked position by the proposed method.

Figure 4.6 Example results of the proposed method dealing with appearance variation and occlusion (2): The center of red '+' means the tracked position by the proposed method.

Figure 4.7 Example results of the proposed method dealing with appearance variation and occlusion (3): The center of red '+' means the tracked position by the proposed method.

Figure 4.8 Example results of FCNT and the proposed method to track the occluded target object. The center of green and red '+' mean the ground-truth and the tracked position, respectively.

Chapter 5

Conclusion

In this paper, we presented the object tracking method which combines SSD, FCN and DP. We confirmed that the proposed method is robust to appearance variation and occlusion through several experiments and achieved the highest accuracy compared to the traditional trackers introduced in the Visual Tracker Benchmark, as shown in Fig 5.1. In contrast to traditional trackers, the proposed method can track the target object without initialization, modifying parameters, and templates as synergies of the combination of SSD, FCN, and DP. Also, the proposed method can be extended to tracking with multiple similar objects by using initialization. We confirmed that using both SSD and FCN is more stable to tracking than single SSD and single FCN as well.

We expect to use the proposed method in analysis field such as traffic analysis, bio-image analysis, etc. In future, we will connect SSD, FCN with network flows [37] to track multi-target objects simultaneously, as shown in Fig 5.2. As we mentioned in Section 4.3.4, we will utilize specific features of a target object to track target object which is occluded by an object of same category, distinguishably. In specific, there are two expected solutions. First is to apply Flownet [38] to utilize information of optical flow.

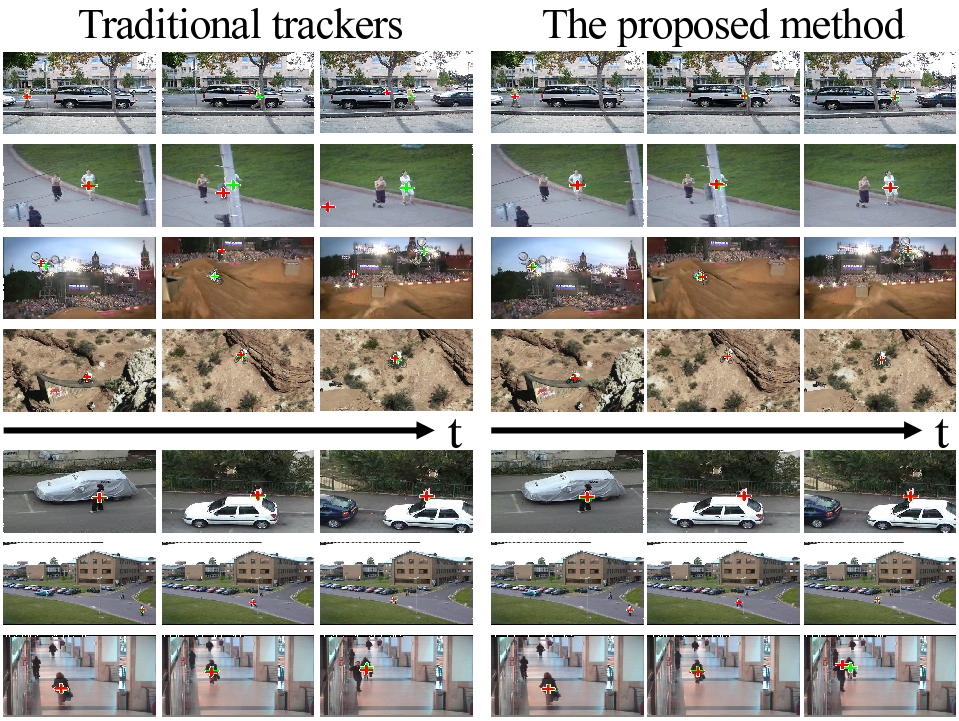

Figure 5.1 Example results of the proposed method dealing with appearance variation and occlusion: The center of green and red '+' mean the ground-truth and the tracked position, respectively. Examples of left side are results of the top five traditional trackers introduced in the Visual Tracker Benchmark. Examples of right side are results of the proposed method.

For the other solution, we will utilize more discriminative features by using features of lower layer of CNN or Network In Network (NIN) [39] or identification neural network [40], additionally.

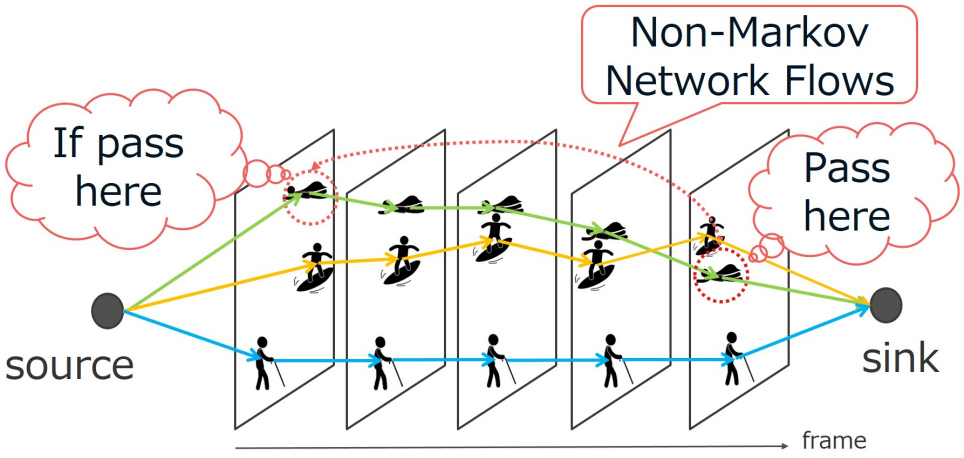

Figure 5.2 Non-Markov Network Flows Model

Acknowledge

First and foremost, appreciate to my supervisor Prof. Uchida. He has been my mentor, and my great friend through the past 2 years. Thanks for him, i spent great time of master course with him and learned not only lots of knowledge but also all things how should i live from now as researcher. Thank you to my supervisor Prof. Uchida again. I also appreciate to Brian Kenji Iwana, Zhu Anna, Shouta Ide, who help a lot for my research. Thank you to all my colleagues of uchida lab family including already graduated senior and the foreign friends who studied with me, my friends and my family.

2018.2

Jinho Lee

Bibliography

[1] Lewis, John P, “Fast template matching,” Vision interface, vol. 95, no. 120123, pp. 15−19, 1995.

[2] Okuma, Kenji and Taleghani, Ali and De Freitas, Nando and Little, James J and Lowe, David G, “A boosted particle filter: Multitarget detection and tracking,” European Conference on Computer Vision, pp. 28−39, 2004.

[3] Comaniciu, Dorin and Ramesh, Visvanathan and Meer, Peter, “Real-time tracking of non-rigid objects using mean shift,” Computer Vision and Pattern Recognition, vol. 2, pp. 142−149, 2000.

[4] Zach, Christopher and Gallup, David and Frahm, Jan-Michael, “Fast gain-adaptive KLT tracking on the GPU,” Computer Vision and Pattern Recognition, pp. 1−7, 2008.

[5] He, Wei and Yamashita, Takayoshi and Lu, Hongtao and Lao, Shihong, “Surf tracking,” Computer Vision IEEE 12th International Conference, pp. 1586−1592, 2009.

[6] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C, “SSD: Single shot multibox detector,” European Conference on Computer Vision, pp. 21−37, 2016.

[7] Long, Jonathan and Shelhamer, Evan and Darrell, Trevor, “Fully convolutional networks for semantic segmentation,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431−3440, 2015.

[8] Uchida, Seiichi and Sakoe, Hiroaki, “A monotonic and continuous two-dimensional warping based on dynamic programming,” International Conference on Pattern Recognition, vol. 1, pp. 521−524, 1998.

[9] Geiger, Davi and Gupta, Alok and Costa, Luiz A. and Vlontzos, John, “Dynamic programming for detecting, tracking, and matching deformable contours,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 3, pp. 294−302, 1995.

[10] Arnold, James and Shaw, SW and Pasternack, HENRI, “Efficient target tracking using dynamic programming,” IEEE Transactions on Aerospace and Electronic Systems, vol. 29, no. 1, pp. 44−56, 1993.

[11] Wu, Yi and Lim, Jongwoo and Yang, Ming-Hsuan, “Online object tracking: A benchmark,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2411−2418, 2013.

[12] Mei, Xue and Ling, Haibin, “Robust visual tracking using $l1$ minimization,” Computer Vision IEEE 12th International Conference, pp.1436−1443, 2009.

[13] Zhang, Tianzhu and Ghanem, Bernard and Liu, Si and Ahuja, Narendra, “Robust visual tracking via multi-task sparse learning,” Computer Vision and Pattern Recognition (CVPR), pp. 2042−2049, 2012.

[14] Han, Bohyung and Comaniciu, Dorin and Zhu, Ying and Davis, Larry S, “Sequential kernel density approximation and its application to real-time visual tracking,”

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1186−1197, 2008.

[15] Jepson, Allan D and Fleet, David J and El-Maraghi, Thomas F, “Robust online appearance models for visual tracking,” IEEE transactions on pattern analysis and machine intelligence, vol. 25, no. 10, pp. 1296−1311, 2003.

[16] Ross, David A and Lim, Jongwoo and Lin, Ruei-Sung and Yang, Ming-Hsuan, “Incremental learning for robust visual tracking,” International Journal of Computer Vision, vol. 77, no. 1, pp. 125−141, 2008.

[17] Kalal, Zdenek and Mikolajczyk, Krystian and Matas, Jiri, “Tracking-learning detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409−1422, 2012.

[18] Grabner, Helmut and Grabner, Michael and Bischof, Horst, “Real-time tracking via on-line boosting,” BMVC, vol. 1, no. 5, pp. 6, 2006.

[19] Grabner, Helmut and Leistner, Christian and Bischof, Horst, “Semi-supervised on-line boosting for robust tracking,” ECCV, pp. 234−247, 2008.

[20] Son, Jeany and Jung, Ilchae and Park, Kayoung and Han, Bohyung, “Tracking-by segmentation with online gradient boosting decision tree,” Proceedings of the IEEE International Conference on Computer Vision, pp. 3056−3064, 2015.

[21] Lawrence, Steve and Giles, C Lee and Tsoi, Ah Chung and Back, Andrew D, “Face recognition: A convolutional neural-network approach,” IEEE transactions on neural networks, vol. 8, no. 1, pp. 98−113, 1997.

[22] Ciresan, Dan Claudiu and Meier, Ueli and Gambardella, Luca Maria and Schmidhuber, Jurgen, “Convolutional neural network committees for handwritten character classification,” Document Analysis and Recognition (ICDAR), pp. 1135−1139, 2011.

[23] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, pp. 1097−1105, 2012.

[24] Sainath, Tara N and Mohamed, Abdel-rahman and Kingsbury, Brian and Ramabhadran, Bhuvana, “Deep convolutional neural networks for LVCSR,” Acoustics, speech and signal processing (ICASSP), pp. 8614−8618, 2013.

[25] Fan, Jialue and Xu, Wei and Wu, Ying and Gong, Yihong, “Human tracking using convolutional neural networks,” IEEE Transactions on Neural Networks, vol. 21, no. 10, pp. 1610−1623, 2010.

[26] Maung, Tin Hninn Hninn, “Real-time hand tracking and gesture recognition system using neural networks,” World Academy of Science, Engineering and Technology, vol. 50, pp. 466−470, 2009.

[27] Torricelli, Diego and Conforto, Silvia and Schmid, Maurizio and D’Alessio, Tommaso, “A neural-based remote eye gaze tracker under natural head motion,” Computer methods and programs in biomedicine, vol. 92, no. 1, pp. 66−78, 2008.

[28] Li, Hanxi and Li, Yi and Porikli, Fatih, “Robust Online Visual Tracking with a Single Convolutional Neural Network,” Asian Conference on Computer Vision, pp. 194−209, 2014.

[29] Wang, Lijun and Ouyang, Wanli and Wang, Xiaogang and Lu, Huchuan, “Visual

tracking with fully convolutional networks," Proceedings of the IEEE International Conference on Computer Vision, pp. 3119−3127, 2015.

[30] Simonyan, Karen and Zisserman, Andrew, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

[31] Everingham, Mark and Eslami, SM Ali and Van Gool, Luc and Williams, Christopher KI and Winn, John and Zisserman, Andrew, "The pascal visual object classes challenge: A retrospective," International Journal of Computer Vision, vol. 111, no. 1, pp. 98−136, 2015.

[32] Hare, Sam and Golodetz, Stuart and Saffari, Amir and Vineet, Vibhav and Cheng, Ming-Ming and Hicks, Stephen L and Torr, Philip HS, "Structured output tracking with kernels," IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 10, pp. 2096−2109, 2016.

[33] Zhong, Wei and Lu, Huchuan and Yang, Ming-Hsuan, "Robust object tracking via sparsity-based collaborative model," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1838−1845, 2012.

[34] Kalal, Zdenek and Matas, Jiri and Mikolajczyk, Krystian, "Pn learning: Bootstrapping binary classifiers by structural constraints," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 49−56, 2010.

[35] Dinh, Thang Ba and Vo, Nam and Medioni, Gérard, "Context tracker: Exploring supporters and distracters in unconstrained environments," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1177−1184, 2011.

[36] Kwon, Junseok and Lee, Kyoung Mu, "Visual tracking decomposition," Computer Vision and Pattern Recognition (CVPR), pp. 1269−1276, 2010.

[37] Zhang, Li and Li, Yuan and Nevatia, Ramakant, “Global data association for multi-object tracking using network flows,” Computer Vision and Pattern Recognition, pp. 1−8, 2008.

[38] Dosovitskiy, Alexey and Fischer, Philipp and Ilg, Eddy and Hausser, Philip and Hazirbas, Caner and Golkov, Vladimir and van der Smagt, Patrick and Cremers, Daniel and Brox, Thomas, “Flownet: Learning optical flow with convolutional networks,” IEEE International Conference on Computer Vision, pp. 2758−2766, 2015.

[39] Lin, Min and Chen, Qiang and Yan, Shuicheng, “Network in network,” arXiv preprint arXiv:1312.4400, 2013.

[40] Li, Wei and Zhao, Rui and Xiao, Tong and Wang, Xiaogang, “Deepreid: Deep filter pairing neural network for person re-identification,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 152−159, 2014.