Robust Template Matching Using Scale-Adaptive Deep Convolutional Features

Jonghee Kim, Jinsu Kim, Seokeon Choi, Muhammad Abul Hasan, and Changick Kim
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
E-mail: {jonghee.kim, jinsu86, seokeon, hasandoesit, changick}@kaist.ac.kr Tel: +82-42-350-7521

Abstract—**In this paper, we propose a deep convolutional feature-based robust and efficient template matching method. The originality of the proposed method is that it is based on a scale-adaptive feature extraction approach. This approach is influenced by an observation that each layer in a CNN represents a different level of deep features of the actual image contents. In order to keep the features scalable, we extract deep feature vectors of the template and the input image adaptively from a layer of a CNN. By using such scalable and deep representation of the image contents, we attempt to solve the template matching by measuring the similarity between the features of the template and the input image using an efficient similarity measuring technique called normalized cross-correlation (NCC). Using NCC helps in avoiding redundant computations of adjacent patches caused by the sliding window approach. As a result, the proposed method achieves state-of-the-art template matching performance and lowers the computational cost significantly than the state-of-the-art methods in the literature.**

I. Introduction

Template matching is considered as one of the core tasks in computer vision as it is the basis of finding solutions to many correspondence identification problems, *e.g.,* visual tracking, object detection, and 3D reconstruction. Template matching is usually performed using a sliding window manner, *i.e.,* all possible patches in an image are compared with the template by using a similarity measuring method. Early template matching methods employ similarity measures such as sum of squared differences (SSD), sum of absolute differences (SAD), and normalized cross-correlation (NCC) [1]. Although template matching methods using such similarity measures work fast, such methods fail to show the robustness against object deformations and partial occlusions.

Using a different trend, several methods attempted to increase robustness with computationally expensive measures. In [2], for deformation invariance, histogram matching (HM) method is proposed which compares the color histogram of the template with the target image patches, without considering the spatial information. Although the method easily deals with geometric deformations, it is still difficult to handle partial occlusion and clutter backgrounds without considering the geometrical cue. In the same vein, Oron *et al.* [3] employ the Earth Mover's Distance (EMD) between two sets of points in $xyRGB$ space. The inclusion of geometrical information xy allows partial occlusion and clutter backgrounds to be handled better than HM based method. In [4], a bidirectional similarity based Best Buddies Similarity (BBS) method is proposed. The BBS is a parameter-free and robust similarity measure method between two sets of points. However, all the mentioned methods require more expensive computations for each window than the traditional similarity measuring methods *i.e.,* SSD, SAD, and NCC.

With the recent success of deep learning algorithms, CNNs are actively employed in a variety of applications *e.g.,* classification, object detection, and segmentation. In [5], CNN is used in an image correspondence problem having a variant of a two-stream network called Siamese network. The Siamese network considers the image correspondence problem as a classification problem which determines the matching similarity between two given patches. Thus, all patches in an image are captured using a sliding window and compared to the template using the Siamese network. For each image patch, exhaustive feature extraction is performed, without considering the redundancies within the image. As a result, the total cost of the computation increases proportional to the number of pixels within the image. Additionally, a large number of parameters are to be trained from the scratch for comparison purposes. Moreover, each input patch is scaled to a predetermined size and fed to CNNs. To this end, CNNs should be learned with a scale-invariant property, which makes training difficulty high.

The above mentioned solutions to the template matching problem suffers from their own limitations. In this paper, we propose a robust and efficient solution to the template matching problem by overcoming those limitations. By an observation that each layer in a CNN represents a different level of deep features of the actual image contents [9], we extract scale-adaptive deep convolutional feature vectors from the template and the input image via the pre-trained VGG-Net [7]. Then, NCC is used to measure the distance between the features of the template and the image to detect the target image patch. Concretely, we propose a scale-adaptive approach which extracts features from an adaptively determined layers of CNNs considering the size of the given template. Figure 1 shows an illustrative diagram of the proposed scale-adaptive deep convolutional feature based template matching method.

The rest of the paper is organized as follows. In Section II, we explain the proposed method in detail. Section III presents the experimental results. Finally, the conclusion of this paper is drawn in Section IV.

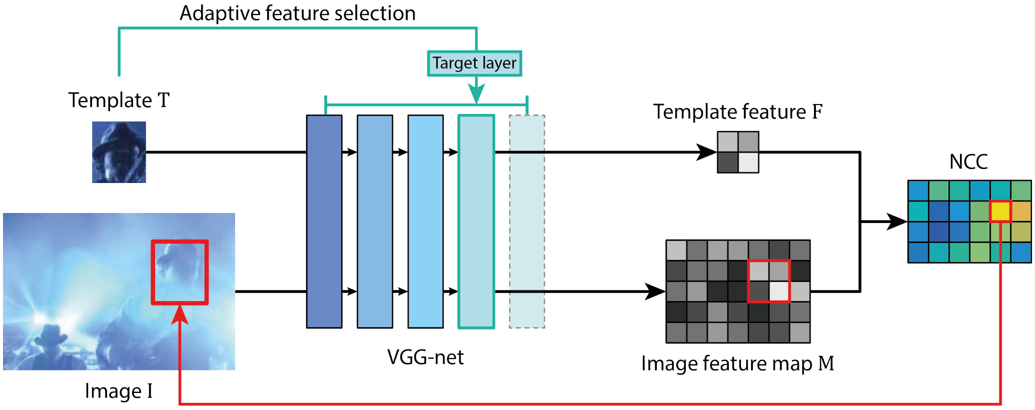


Fig. 1: Illustration of the scale-adaptive deep convolutional feature extraction based method for template matching.

II. Proposed Method

We define template matching as a problem of locating the most similar patch within an image $I \in R^{m\times n\times 3}$ for a given template $T \in R^{w\times h\times 3}$, where m and n represent the width and height of the image and w and h correspond to the width and height of the template. To solve the problem, we propose a robust and efficient scale-adaptive deep convolutional feature based method. The detail of the proposed method appears in the following subsections.

A. Scale Adaptive Deep Convolutional Feature Extraction

In the proposed scale-adaptive feature extraction method, we use the VGG-Net [7] to extract feature vectors from both the template and the input image. Unlike common CNNs-based methods, we do not scale the template or images into a specific size, *e.g.*, 224×224. Instead of scaling, taking the template size into account, we adaptively identify the target layer of the VGG-Net and extract the feature vectors from the target layer. Each layer of CNNs has a $rf^l \times rf^l$ receptive field where a width of a receptive field of the lth layer rf^l is as defined as

$$rf^l = \begin{cases} rf^{l-1} + (f^l - 1)\prod_{i=1}^{l-1} s^i & l > 1, \\ 3 & l = 1, \end{cases} \tag{1}$$

where, for simplicity, the layer index l is set depending on their order, *e.g.*, conv_{1_1} is 1, conv_{1_2} is 2, and pool_1 is 3. The f^l represents a filter size of the lth layer and s^i represents a stride of the ith layer. If a template is smaller than a receptive field of the target layer, the layer deals with a meaningless outer region of the template filled with zeros. Therefore, we limit the target layer to have a receptive field which is smaller than or equal to the template. Here, we can represent a target layer index l^* by

$$l^* = \max(l - k, 1) \quad \text{s.t.} \quad rf^l \le \min(w, h), \tag{2}$$

where k should be greater than or equal to 0 in order to satisfy the condition in Eq. (2). We set k to 3 for consistent selection while dealing with templates of various sizes. The effect of the value of k is shown in Section III. As we set k to 3, the size of the receptive field in the $(l-3)$th layer is about half of $\min(w, h)$ because there is a pooling layer between the lth and the $(l-3)$th layers. Then, we feed a template T and an image I into CNNs and extract a template feature F and an image feature map M from the target layer. Notice that an image I and a template T are padded with a few zeros before feed-forwarding in order to avoid generating fractional output. For example, since pool_1 layer has a 6×6 receptive field and a stride of 2, the input should be padded to have a size of $6+2d$ where d is an integer greater or equal than zero. Here, we only need to compute convolutional feature vectors once for each template and image with a fully convolutional approach [6], which is much more efficient than a naïve sliding window approach.

B. NCC-based Similarity Measure

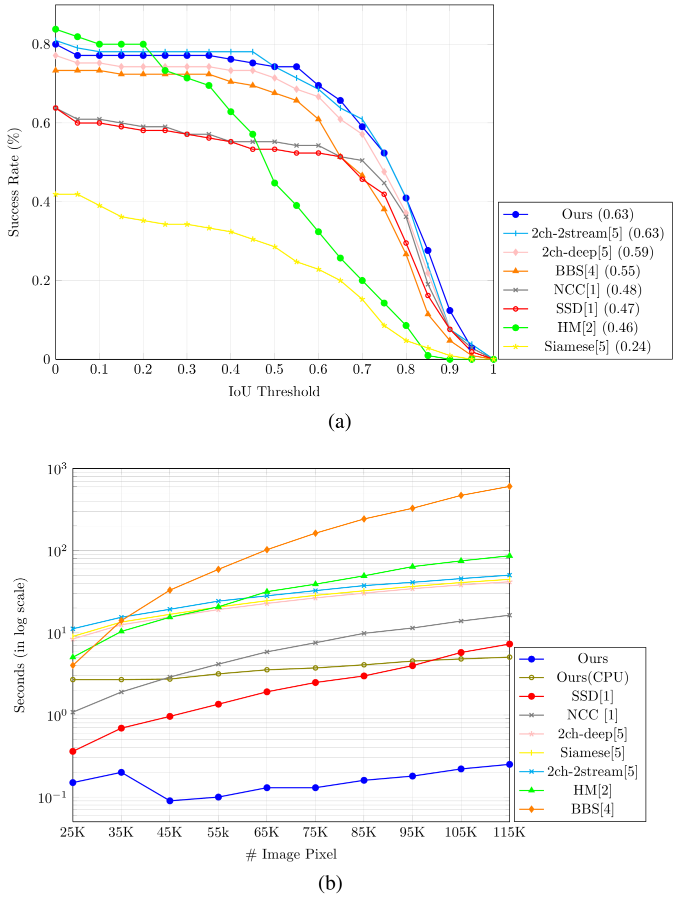
We locate the most similar patch by using NCC between M and F. First, we calculate NCC between M and F by

$$NCC_{i,j} = \frac{< F, \widetilde{M} >}{|F||\widetilde{M}|}, \tag{3}$$

where $\widetilde{M} = M_{i:i+h_f-1, j:j+w_f-1}$, which is a feature patch extracted from M with a width w_f and a height h_f as equal to the size of F. Here, we do not subtract the mean of each F and $\widetilde{M}$ since they have already been rectified. Then, we find the location (i^*, j^*) which has the maximum NCC value. Since the location is found on feature domain, we recover a box position on image domain corresponding to the location by back-projecting the position.

C. Location Refinement

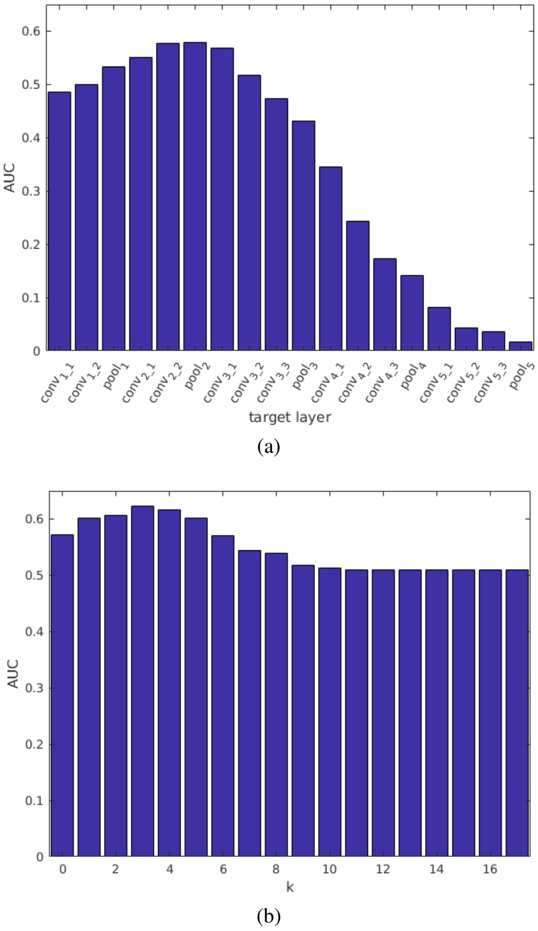
Since the proposed method is conducted on the convolutional feature map, it is considered as performing sliding window with a stride of the target layer. Therefore, we refine the location found on feature domain in order to achieve better precision. First, we set the initial box position $((x_1^0, y_1^0), (x_2^0, y_2^0))$ computed by back-projecting the maximum location (i^*, j^*) into image space where (x_1^0, y_1^0) is the upper-left position



(a)

(b)

Fig. 2: Quantitative analysis of the performances of the proposed method and the state-of-the-art method. (a) Success curve and AUC, (b) Computation time with respect to the image size in terms of pixel.



(a)

(b)

Fig. 3: Performances analysis based on AUC using (a) fixed scale features and (b) the proposed adaptive-scale features.

and (x_2^0, y_2^0) is the bottom-right position of the initial box, respectively. Then, we take patches in the vicinity of the maximum location (i^*, j^*) on NCC for refinement. In detail, to get refined x_1, we employ a weighted sum of the original locations of a 3×4 NCC patch where weights are NCC values expressed by

$$x_1 = \frac{\sum_{u=-1}^{1}\sum_{v=-2}^{1} NCC_{i^*+u,j^*+v}\cdot(x_1^0 + v\cdot\prod_{i=1}^{l^*-1}s^i)}{\sum_{u=-1}^{1}\sum_{v=-2}^{1} NCC_{i^*+u,j^*+v}}. \tag{4}$$

In a similar manner, we also refine x_2, y_1, and y_2.

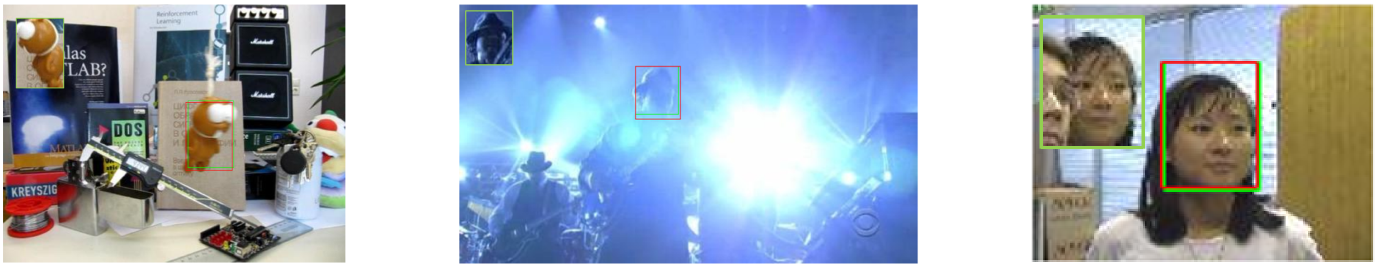
III. Experiments

A. Experiment Setup

In the experiment, we follow the evaluation protocol of [4] for fair comparison. In detail, 105 template-image pairs are sampled from 35 videos (3 pairs per video) from the tracking dataset given in [10]. The template is randomly chosen and the image is sampled 20 frames after the template is captured. For each pair, intersection-over-union (IoU) between ground truth box and predicted box is measured. Then, success curve with varying IoU thresholds and area under the curve (AUC) is used for quantitative comparison. We compare the proposed method to 7 state-of-the-art methods in the literature. The methods are three CNNs-based methods (2ch-deep, 2ch-2stream, Siamese) [5], BBS [4], HM [2], SSD, and NCC [1].

B. Experimental Results

We perform several quantitive analysis to show the effectiveness of the proposed method. First, we perform a quantitative analysis using success curves shown in Fig. 2 (a). As it can be seen, the performance of the proposed method is comparable to that of 2ch-2stream method [5] and performs better than that of 2ch-deep [5] and BBS [4] methods. Although the performance of the 2ch-2stream is comparable, it is due to the fact that the 2ch-2stream method is explicitly designed for measuring the similarity between two patches. In contrast, the proposed method achieves the same performance as 2ch-2stream with the pre-trained VGG-net by taking advantage of the scale-adaptive approach. It is further noticeable in Fig. 2 (a) that the Siamese network [5] shows worse performance than simple similarity based methods such as SSD and NCC even though the method also uses CNNs.



(a) Deformation

(b) Illumination change

(c) Partial occlusion

Fig. 4: Examples of the template matching performance in different challenging situations using the proposed scale-adaptive deep convolutional feature based method. Green and red boxes represent ground truth and a predicted box, respectively. The above images are from the tracking dataset [10] which we used for evaluation.

From the efficiency perspective, we analyze the running time performances of the proposed method and the state-of-the-art methods which are reported in Fig. 2 (b). We compare the running time performances of the methods with respect to the image of size ranging from $25K$ to $115K$ pixels. The computation time of the proposed method primarily depends on the feature extraction step due to a high complexity of the CNN. The methods based on CNNs with accelerated computing using GPUs takes tens of seconds due to exhaustive feature extraction using a sliding window. In contrast, we perform feature extraction once for an image rather than a series of sliding windows. As a result, the proposed method could be as efficient as NCC after feature extraction. Therefore, it takes only 0.2s with GPU acceleration and 5.0s with only CPU, which performs significantly faster than of BBS (605.0s), HM (86.5s), CNNs-based methods (44.7s $\sim$ 50.3s), NCC (16.4s), and SSD (7.3s) when dealing with an image with $115K$ pixels. It is worth noting that the proposed method is faster than even SSD since we calculate NCC with small feature map caused by using a latter layer which has large stride. For instance, a spatial size of feature map from the pool$_5$ layer is 32^2 times smaller than the number of image pixels.

The effectiveness of the proposed scale-adaptive feature is shown in Fig. 3. For the fixed-scale case, we match the template features to the image features extracted from all layers of the VGG-Net except fully-connected layers. The layer pool$_2$, containing a 14×14 receptive field, shows the best performance with AUC 0.57. For scale-adaptive feature extraction, we change the value of k from 0 to 17, corresponding to the layer which had a similar receptive field size of a template and the first layer, conv$_{1\_1}$, respectively. When k is set to 3, it shows the best performance and it is not sensitive to the change in the vicinity of 3. Moreover, it performs better than the best-performing fixed-scale feature extraction method. It verifies that using a scale-adaptive deep convolutional feature based method is more effective than the fixed-scale case. In addition, average of $\min(w,h)$ for templates which we use for evaluation is 40. In the case where $\min(w,h)$ is 40, with $k=3$ made l^* represents pool$_2$ which is the best outcome for fixed feature extraction. This performance also supports the reason of effectiveness of $k=3$.

Next, we compare the proposed method with and without refinement. Without refinement, the proposed method shows AUC of 0.62. And, refinement increased precision in terms of AUC by 0.01. Although the precision increment does not look significant, it helped to deal with the sizes differences between a template and a target object in an image. Figure 4 shows a few qualitative comparisons in different challenging situations. As it can be seen, the proposed method finds the best matches almost as accurately as the labeled ground truths.

IV. Conclusion

In this paper, we have proposed a robust template matching method using scale-adaptive deep convoultional features. The scale-adaptive approach could deal with various sizes of templates by using rich and scalable representations of CNNs properly. In addition, to prevent redundant computations of sliding window based methods, we have employed a fully convolutional approach for feature extraction followed by efficient normalized cross-correlation based location. As a result, the proposed method achieves state-of-the-art performance within affordable time.

References

[1] W. Ouyang, F. Tombari, S. Mattoccia, L. Di Stefano, and W. K. Cham, “Performance evaluation of full search equivalent pattern matching algorithms,” *PAMI*, 2012.

[2] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” *CVPR*, 2000.

[3] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, “Locally orderless tracking,” *IJCV*, 2014.

[4] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman, “Best-buddies similarity for robust template matching,” *CVPR*, 2015.

[5] S. Zagoruyko, N. Komodakis, “Learning to compare image patches via convolutional neural networks,” *CVPR*, 2015.

[6] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional models for semantic segmentation,” *PAMI*, 2016.

[7] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, 2014.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, and N. Zhang, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition,” *ICML*, 2014.

[9] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*, 2014.

[10] Y. Wu, J. Lim, and M. Yang, “Online object tracking: A benchmark,” in *CVPR*, 2013.