IOP Conf. Series: Materials Science and Engineering **435** (2018) 012056 doi:10.1088/1757-899X/435/1/012056

Robust Template Matching via Pruning Deep Feature

F Wang[1], J P Xiong[1], J Yin[1] and X M Zhang[1]

[1]Zhejiang Dahua Technology CO., LTD., Zhejiang Province, Hangzhou, China

Email: wzfeng1991@163.com

Abstract. The application environment with slightly diversity between template image and target image has been the mainstream in template matching over the past decade or so. This paper, however, will discuss template matching method in such scenarios with erratic weather, deformations, scaling etc. For the feature engineering, by choosing the appropriate layers in a CNN and pruning inefficient convolutional kernels in the layers we want, we construct a feature space that can be used to represent more complex features compared to the traditional computer vision feature engineering, such as corners, colours, edges etc. Meanwhile, the feature space and computational capacity can be greatly reduced by the pruning operation on convolutional kernels. For similarity measure, a distance penalty term on the feature of image patches will be added in the final score function to make our method robust to deformation and scaling. Furthermore, the key coefficients of penalty term have been opened so that our method can be adjusted based on the actual scenarios. Numerous experiments on the benchmark dataset are conducted accompanied by comparisons with a few recent proposed methods, e.g., BBS, DDIS, etc. The results have demonstrated well the robustness and accuracy of our method relative to the other methods. Note that, our method can reach the industrial standard with GPU acceleration.

1. Introduction

The study of template matching, as a fundamental issue in the fields of computational photography and computer vision, is always active over the years, and we note that some future developing trends in this study is motivated by the application environment for complex scenarios in industry. The major mission of this work is asked to solve a problem: how to increase the robustness of the template matching algorithm under illumination changes fleetly and background interface mightily, as illustrated in figure 1.

Based on the analysis of existing literature, the emphasis of template matching can be split into two parts[1]: feature engineering and similarity criterion. Feature engineering is the first step of computer vision tasks, and is also the most critical step. There could be the low-level features[2], such as edges, corner[3], which can be comprehended by humans, and the high-level features, such as hand-craft features or more abstract features learned by a CNN(convolutional neural network)[4] with a big training set. Similarity criterion is another research emphasis in template matching, which is used to find the minute location of the template in a big image. As pointed in [5], the most significant performance impact in a computer vision tasks is the quality of feature engineering relative to other modules, like pre-processing, regularization term, numerical optimization. In this work, we redefine the deep feature of a CNN to improve the positioning precision in complex environment, and redesign the calculation method which is more efficient for these high dimensionality feature space. Though template matching and object detection are two themes in the computer vision, they have some

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

internal connections[6]. Both of them need to find the specific location of an object in a big image. The number of objects is settled in object detection, and a template in the work procedure of template matching could be just one of these objects. However, the feature engineering in template matching is consistent with object detection in essence. Inspired by the great success of deep learning achieved in object detection, such as Faster RCNN[7], YOLO[8], etc., we redefine and prune the basic convolutional network that make it applicable to template matching. Specifically, the descriptive ability of inter-layers or kernels in one layer will be analyzed in detail here. We find that the features described by the previous layers in VGG-Net[9] have a low-level characteristic, such as large edge, colour information, which can be comprehended by mankind. Whereas the last layers in VGG-Net[9] could describe more abstract features which contain distinct semantic information. These features with apparent semantic information may really suit object detection, but are not suitable for building the feature engineering in template matching and will be counterproductive in many cases. How intricate the feature is applied in a computer vision technique hinge on the specific case. Meanwhile, the kernels in one layer of a network structure appear in an endless variety of peculiarity. We can see through our studying and analysing, that twenty per cent of the kernels concentrate the vast majority energy in one layer. Those low-strength convolutional kernels are inadaptable to the problem of template matching. And, it means that we just need ten to twenty per cent of the convolutional kernels in one layer to construct the feature engineering. By designing the feature in this way, we can construct an more efficient feature engineering with lower storage space requirement and higher computation efficiency.

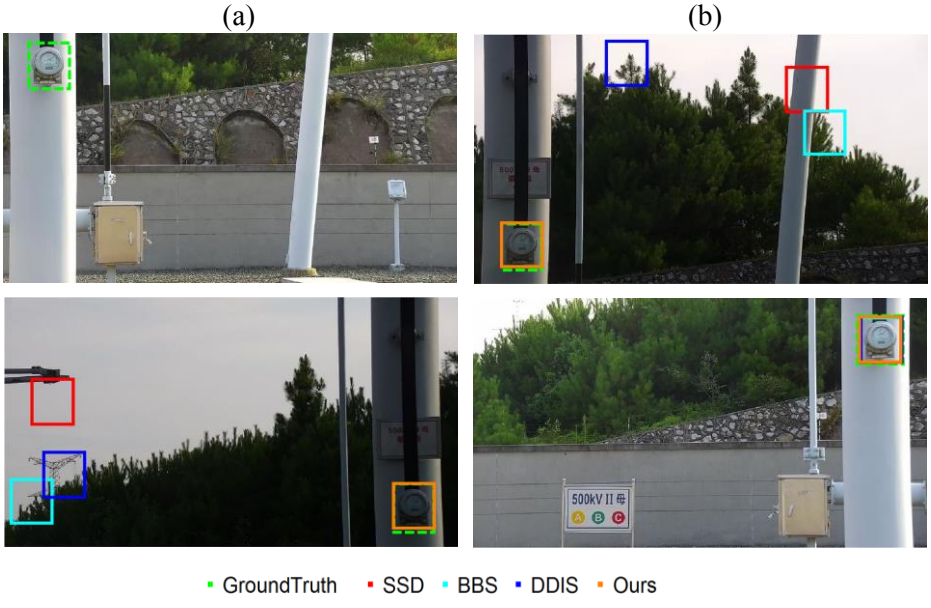


Figure 1. Template Matching under illumination changes and background interface: Template matching expression of different methods. (a). The dial plate in this image is marked as template (in green). (b)(c)(d), the performance of different template matching are marked in these complex luminance images.

As in the earlier study, comparing the template and the target image pixel by pixel is the most common approach used in similarity criterion. Nevertheless, such a mode can't handle robustly with the deformation[10]. To solve this problem, the features extracted by some algorithms will be mapped into a higher dimensional space in [11], meanwhile, the number of nearest neighbours will be obtained to establish a heat map[12]. The specific location of the template will be confirmed by this heat map. The feature engineering used in BBS[11] is low-level, and its' computational power is highly inefficient, in particular for a 1080p HD target image. The computing time will be spent 20 minutes or more with a hardware configuration of Intel Core i5, 8G memory. Although the problem about

calculation efficiency has been solved by DDIS[13], it still can't work well under illumination changes fleetly and background interface mightily.

The application environments of template matching in the industry are indoors mostly with a good light condition. But we will mainly discuss template matching used in the outdoors with complicated scenarios, such as unpredictable weather, complex background and various industrial components. Meanwhile, the deformation and scaling partly are considered as well. Our contributions can be summarized as follows:

Firstly, we construct the feature engineering used in template matching with a pruning operation on CNN. The pruning operation mainly contains two parts. For the inter-layers in CNN, the ahead and middle layers that can extract texture or colour information of the image will be applied to final feature space. And the last layers with obvious semantic information have been discarded here. For convolutional kernels in inter-layers, more than eighty per cent of kernels will be pruned to optimize the feature space and accelerate the calculations.

Secondly, in order to overcome the unforeseeable elements in industry, such as deformation, scaling etc., a new similarity measure with adjustable coefficients based on the actual situations has been proposed here. For the high-dimensional feature space constructed by deep features, we use the product quantization based nearest neighbour method to solve this problem.

2. Proposed Framework

In this section, we will discuss how to prune the inter-layers and intra-layers in a CNN and how to design the similarity measure to strengthen the robust aimed at deformation and scaling.

2.1. Pruning Deep Features in Template Matching

2.1.1. Feature Description of Different Inter-Layers

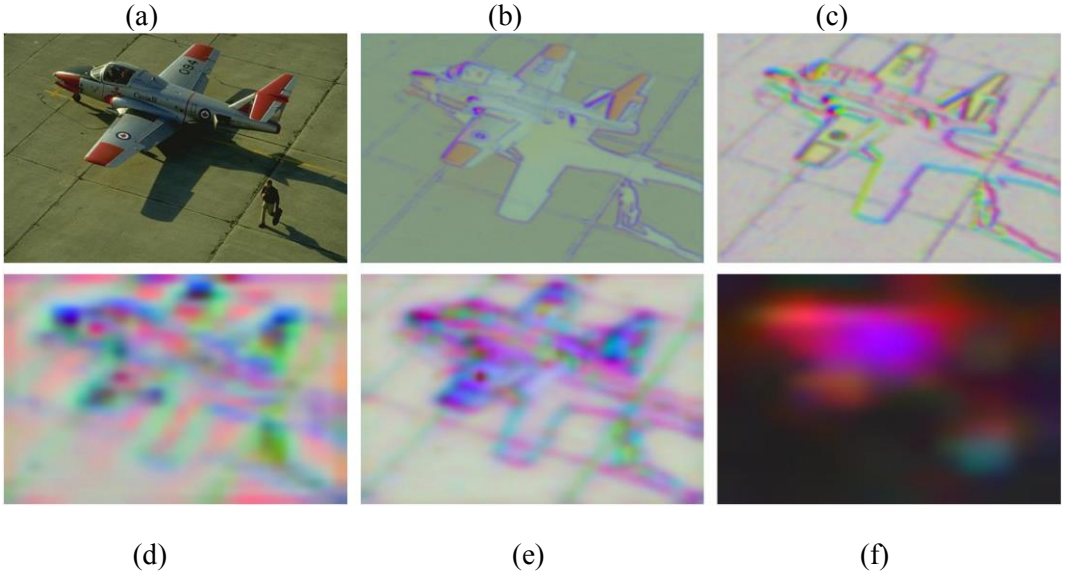


Figure 2. Visual comparison of different layers' intermediate results in VGG-19 for the 'trainer aircraft' image from 'BSD500' dataset. (a) is the original image. (b)-(f) reveal the feature map extracted from convolutional layer *conv1_2*, *relu2_2*, *conv3_4*, *relu4_4* and *conv5_1*.

Many pre-training convolutional neural network that involved in some object detection algorithms, such as VGG-Net[9] pre-trained by the large-scale ImageNet dataset[14], possess the ability to generate different levels of features in an input image. Taking VGG-Net[9] as an example, the

distinction between features extracted from different layers will be discussed in detail here. The other networks, e.g. AlexNet[15], ResNet[16], can also be used in template matching.

With the increase of convolutional layers and pooling layers, the spatial resolution of the feature space produced by different layers in VGG-19 will be decrease in proportion. For example, the spatial size of feature maps on the *conv5_1* will be decrease to 7×7 in the VGG-Net[9], which is $1/32$ of the input image size 224×224, due to the pooling operation. In order to make these layers' output have the identical contribution to the final feature engineering, all these output features will be resized to the input image size by bicubic interpolation.

Figure 2 demonstrates visible feature maps of different layers in VGG-Net[9] pre-trained by the large-scale ImageNet dataset[14]. The shallow layers' feature description corresponding to figure 2 (b) and figure 2 (c) reserve abundant edge structure details and color information, and have good recognition ability to area with high contrast. These low-level feature maps are very close to the feature engineering produced by theoretical analysis or handcraft design, which is efficient for locating the template. Nevertheless, the feature maps will be shrunken and gather together to form more high-level abstract feature for the final layers of VGG-Net[9]. The feature map shown in figure 2 (f) reveals the specific locations of the training plane and the pilot, which exhibits the semantic information of templates obviously. These high-level feature maps with semantic information will be indispensable in the task of objection detection, image classification etc. However, the result, shown in the next section, reveals that these high-level features are unnecessary for template matching, and even might have negative impact on the feature engineering. The negative impact can be attributed to two aspects. On one hand, fine-grained features will be more useful for template matching, relative to the high-level feature with semantic information. On the other hand, because the basic network VGG-Net[9] used here is pre-trained by the large-scale ImageNet dataset[14] with category-level 1000 labels, the final feature maps will be invalid once the object in the template does not belong to the 1000 labels, which is obviously shown in figure 2 (f).

According to the above comprehensive analysis, the layers that could extract the fine-grained features, such as edge, color, etc. should be used to construct the feature engineering used in template matching, instead of the layers with high-level semantic characteristics.

2.1.2. Feature Description of Kernels in Intra-Layer. There are great redundancy kernels in the pre-trained VGG-Net[9] by observing, which are inadaptable to construct the feature space in template matching. In general, the smaller the magnitude of the convolutional kernel, the lower the intensity of the response to feature – the majority of these kernels' response will be decay to become a 'zombie' [17], which cannot be used to express the image edge of color information.

The effectiveness of convolutional kernels in CNN can be measured by the intensity of kernels' response, which can be expressed numerically as sum of absolute value. The convolutional kernel is noted as $\boldsymbol{K}\in R^{i\times j}$ (e.g. 3×3), and the magnitude of $\boldsymbol{K}$ is noted as $k_{i,j}$. And we measure the intensity of convolutional kernel as

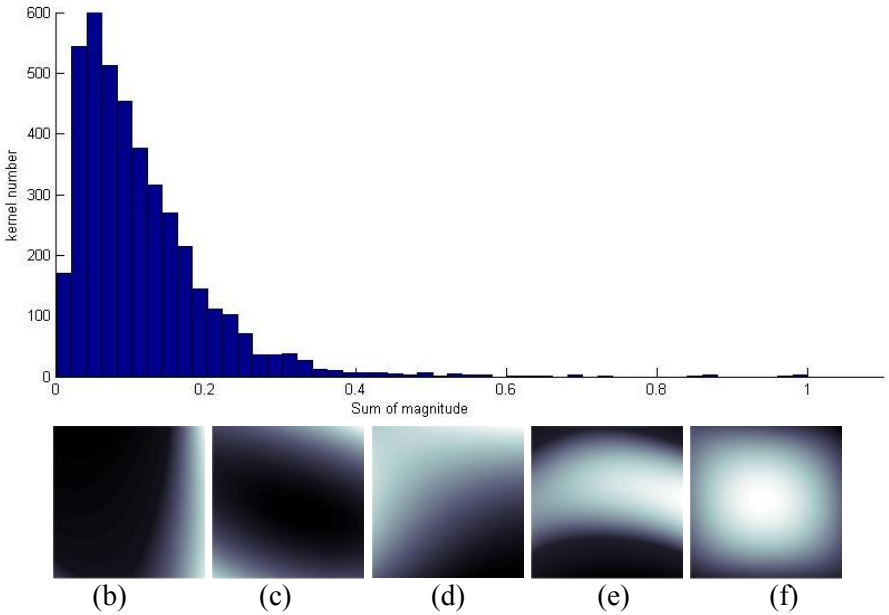
$$S=\sum\left|k_{i,j}\right| \tag{1}$$

The intensity response statistics of 64×64 kernels in *conv2_1* layers of VGG-Net[9] is shown in figure 3, which have been normalized to 1. For a better visualization, kernels with intensity response 0, 0.2, 0.4, 0.6, 0.8, 1 are interpolated as shown on the second line of figure 2.

It can be observed that convolutional kernels in VGG-Net[9] blur image (for reducing details, similar to a normal distribution as shown in figure 3(f)) or sharpen image (for emphasizing the details or directions of object as shown in figure 3(e)). Meanwhile, there are plenty of 'zombie' kernels, as shown in figure 3 (b), (c), which have a low response to pattern information, such as edge, color. The results reveal that the proportions of kernels with strength less than 0.2 and 0.4 account for 79.91% and 97.36% respectively. These 'zombie' kernels will weaken predominantly valid features' influence

in the engineering feature used in template matching, and augment the time complexity and the space complexity.

(a)



(b) (c) (d) (e) (f)

Figure 3. Statistical and visualization of kernels corresponding to *conv1_2* layer in the VGG-Net. (a) shows the statictics of intensity response, and (b)-(f) is interpolated visually for kernels with intensity response 0, 0.2, 0.4 ,0.6, 0.8, 1.

The experimental results showed that the majority of kernels in intra-layer are unnecessary for feature space used in template matching. By removing these 'zombie' kernels, we can improve the localization effect and reduce calculation amount which is especially effective in terms of similarity measure.

2.2. Similarity measure in NN-based Space

Plenty of location methods have been applied here, such as affine mapping[18][19], tone mapping[20]. An approximate solution, which decides whether the template and the target image window possess the same distribution, is elaborated here as a similarity measure[11]. Meanwhile, in order to cope with complex situation, a deformation penalty term is added, and a new searching strategy is adopted for the complex deep feature space constructed above.

The patches of the template image and the target image window is written as t_i , $s_j \in R^d$. Hence, the object function is measure the similarity between these two sets $T = \{t_i\}_{i=1}^N$ and $S = \{s_j\}_{j=1}^N$ in nearest neighbor space, where T denotes template image and S denotes target window image. Both experimental results and theoretic analysis indicate that the unique nearest neighbor of $t_i \in R^d$ can be found in $s_j \in R^d$ when the template image and target window image have the same or similar distributions. We take the feature patches of template and target window image as $t_i^f \in R^d$ and $s_j^f \in R^d$, so we define the nearest neighbor of t_i in S^f as:

$$s_j^f = \arg\min_{s \in S^f} dis(t_i^f, s) \tag{2}$$

where $dis(t_i^f, s)$ is distance calculation formula of feature sets, such as l_1 or l_2 norm, and the total number of nearest neighbor of t_i^f in s_j^f is noted as $num(t_i, S^f)$. As t_i^f with the least nearest neighbor in S^f should have a greater weight, we adopt the reciprocal form of $num(t_i, S^f)$ in the final score function.

The practical engineering, template matching should be robust to deformation since the target objects often have shifted or rotated in reality, which is reflected as some jitter in feature space. Here, we note the location information of patches in template and target image as t_i^l, s_j^l respectively, and the distance between them is measured as $d = dis(t_i^l, s_j^l)$ which can be the Euclidean distance. So the penalty term of distance in the final similarity score is shown as follows:

$$Penalty(d) = a^{b^d} \tag{3}$$

where $a > 0$ and $b > 1$, which is decided concretely by the degree of deformation in reality. Specifically, the penalty term will restrain the similarity score when $d = dis(t_i^l, s_j^l)$ is large than a certain threshold. Hence, the penalty strategy is applied to (2), leading to a general score function as follows:

$$Score(T, S) = \sum_{t_i \in R^d} \frac{1}{num(t_i, S)} a^{b^d} \tag{4}$$

A similarity score map can be obtained by calculating score between template and image sliding window in the target image with formula (4), and the final template's location in target image can be determined by the highest scores in the score map.

The feature space constructed by numerous convolutional kernels of a CNNs can't be solved effectively by the conventional brute-force, KD-tree search strategy[21] for approximate nearest neighbor. Hence, product quantization based nearest neighbor method[22] is adopted here. In this method, the feature space will be divided into several subspaces by building codebooks and there is new searching strategy for it. After quickly positioning the certain subspace, the nearest neighbor feature we want can be obtained by traversal operation[23].

3. Experimental Results

In this section, we will discuss the performance of our method and make comparisons to the classical methods (SSD[1]), state-of-the-art methods (BBS[11], DDIS[13]). Meanwhile, an outdoor testing dataset is constructed here to evaluate the performance of these algorithms.

3.1. Data Preparation and Parameter Settings

A testing dataset to the template matching for outdoors is constructed here, and the ground truth location of template object in the test images is marked by professional workers. The entire data collections contain 1861 scenarios, and a template corresponds to three different testing images in different scenarios, including the change of weather, deformation, scaling etc. as illustrated in figure 1. All these high-resolution images with 1920×1080 format are captured by industrial camera. In the testing process, the ground truth in the first image, such as figure 1(a) will be treated as template, the other three images in figure 1 are the target images.

To note that, the experiment results of BBS[11] and DDIS[13] with default arguments are produced with codes from the homepage of their authors, and SSD[1] is implemented by ourselves. And the parameters of our methods mainly comprise three components:

1. Inter-layers: the shallow layers of CNN will be used here, such as *conv1_2*, *relu2_2*, *conv3_4* in VGG-Net.

2. Intra-layers: 10% of the convolutional kernels in a layer will be kept according to the strength of these kernels' response discussed in section two.
3. Similarity measure: the coefficients 'a' and 'b' in formula (4) are set to 1.09, 1.22 respectively for the scenarios with slight deformation or scaling.

IoU (Intersection-over-Union)[24] is utilized in our approach felicitously as an evaluation criteria. For the complicated scenarios, the location with IoU greater than 80% is to be regarded correct, and we will calculate accuracy of these template matching methods according to this stipulation.

3.2. Results and Comparison

Templates in the testing dataset can be various, such as instruments, character etc. figure 2 demonstrates the template matching results in the case of varying illumination. We can see that the complicated background including trees, telegraph pole can also cause large perturbations to the matching results.

The matching object in figure 4 is a Chinese character as shown in figure 4 (a), and the target images' background is complicated. For one thing, the matching process will become especially complicated with the different weather conditions, such as cloudy sky, clear sky, which is hardly modeled mathematically. For another, some unexpected watermarks might be encountered due to the industrial applications or emergency situations. These factors have negative effects on the robustness of the proposed algorithm. As shown in figure 4, the different weather conditions between (a) and (c) directly lead to the failure using SSD[1], BBS[11], and some watermarks lead DDIS[13] out of work. Many experimental results here explain that the huge gap between these template matching methods is mostly caused by the feature engineering used. The features used in conventional computer vision including corners, edge, colors don't possess the ability to describe the complicated scenarios, such as cloud in the sky, the complex Chinese character.

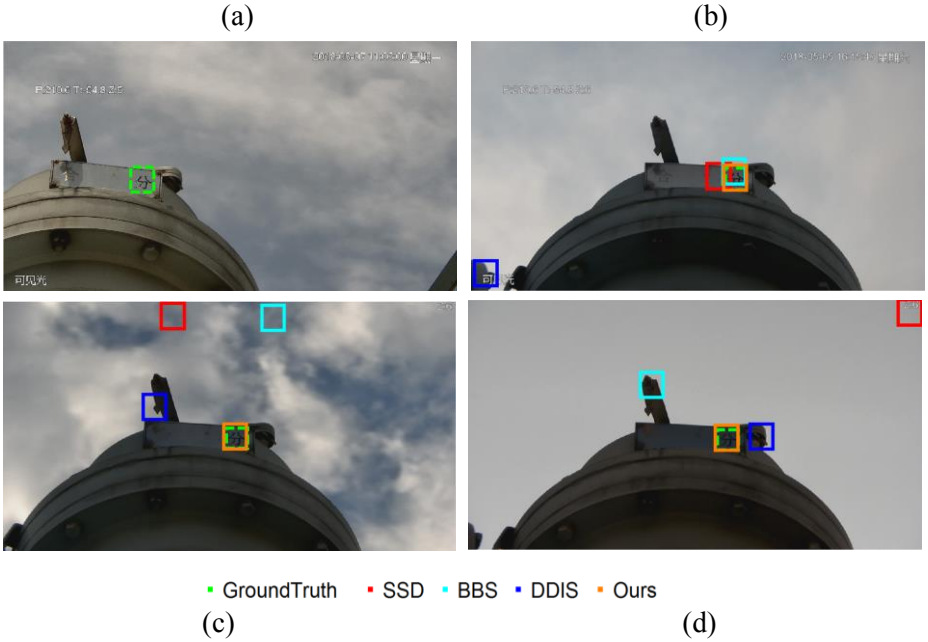


Figure 4. The template matching results of a Chinese character under illumination changes and background interface.

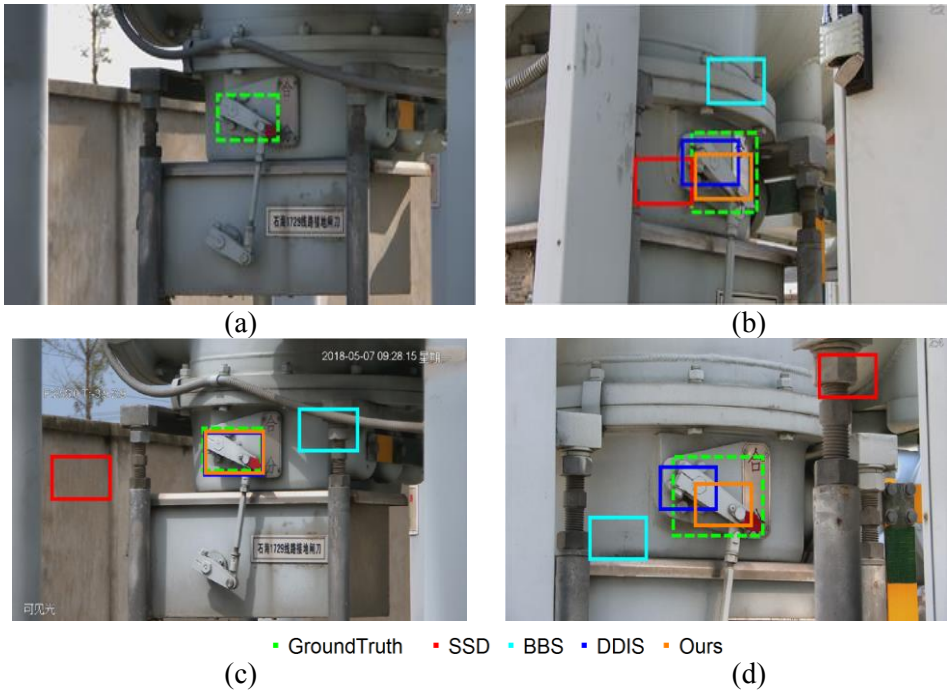


Figure 5. The template matching results of object with deformation and scaling under illumination changes and background interface.

The object template in target images with deformations and scaling is demonstrated figure 5. SSD[1], and BBS[11] failed in this case thoroughly mostly attributes to the illumination, deformation, scaling. Although both DDIS[13] and our method can locate the template in the target images, the localization effect of our method is more effective than DDIS [13] under the standard of IoU.

Table 1 provides the average accuracy and time-consuming of SSD[1], BBS[11], DDIS[13] and our method under the assessment criteria we make above. And our testing environment is a computer configured to I5 processor, 8G RAM, Windows 7 Ultimate, Matlab 2014a. SSD[1] and BBS[11] may not suit the complicated outdoors with an accuracy of 48.9% and 61.7% respectively. And DDIS[13] shows a notable improvements in accuracy compared to BBS[11]. Some of the scenarios that cannot be solved by our method mainly attributes to the illuminations change sharply relative to template image to the extent that also cannot be identified by human. In terms of time-consuming, BBS[11] is the slowest, and the run time of our method would take 23.7s averagely. However, we can use the graphics processor (GPU) to accelerate our method so that it can satisfy the requirements of applications in reality.

Table 1. The accuracy and time-consuming of different method.

Method	SSD	BBS	DDIS	Ours
Accuracy	48.9%	61.7%	82.1%	98.7%
Time (s)	1.2	1308.1	9.5	23.7

4. Conclusions

This paper proposes a high-efficiency template matching method to solve the complicated scenarios with illumination variation, deformation, scaling etc. with a pruning operation on CNN and new similarity criterion. The pruning operation contains two parts. On the one hand, within the template matching region we select the ahead and middle partially layers of a CNN to construct the feature engineering according to their character. And only 10% around convolutional kernels in a layer of CNN is used in the final feature space because more than 80% kernels' strength of response to the images' details could be canceled out. For the similarity measure, an adjustable parameter penalty

term is proposed here to make our method robust to the deformation and scaling. The final experimental results show that under the scenarios with illumination variation, deformation, scaling, a good matching result can be gained with our method compared to the other template matching method, such as BBS[11], DDIS[13]. The follow-up will explore more complicated scenarios with noise[25], or improve efficiency of the algorithm[26] to make it more accommodative in industry.

References

[1] Ouyang W, Tombari F, Mattoccia S, Di Stefano L and Cham W K, Performance evaluation of full search equivalent pattern matching algorithms, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol **4**, no. 1, pp. 127-143, 2012.

[2] Balntas V, Lenc K, Vedaldi A and Mikolajczyk K, HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp.3852-3861, 2017.

[3] Rublee E, Rabaud V, Konolige K and Bradski G, ORB: An efficient alternative to SIFT or SURF, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2564-2571, 2011.

[4] Schmidhuber J, Deep learning in neural networks: An overview, *Neural networks*, vol. **61**, pp. 85-117, 2015.

[5] Wang N, Shi J, Yeung D Y and Jia J, Understanding and diagnosing visual tracking systems, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3101-3109, 2015.

[6] Felzenszwalb P F, Girshick R B, McAllester D and Ramanan D, Object detection with discriminatively trained part-based models, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. **32**, no. 9, pp. 1627-1645, 2010.

[7] Ren S, He K, Girshick R and Sun, J., Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. **39**, no. 6, pp. 1137-1149, 2015.

[8] Redmon J, Divvala S, Girshick R and Farhadi A, You only look once: Unified, real-time object detection, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 779-788, 2016.

[9] Simonyan K and Zisserman A, Very deep convolutional networks for large-scale image recognition, Computer Science, *IEEE Int. Conf. Learning Representation (ICLR)*, 2015.

[10] Kim H Y and De Araújo S A, Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast, *In Pacific-Rim Symposium on Image and Video Technology*, pp. 100-113, 2007.

[11] Dekel T, Oron S, Rubinstein M, Avidan S and Freeman W T, Best-buddies similarity for robust template matching, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2021-2029, 2015.

[12] Simakov D, Caspi Y, Shechtman E and Irani M, Summarizing visual data using bidirectional similarity, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1-8, 2008.

[13] Talmi I, Mechrez R and Zelnik-Manor L, Template matching with deformable diversity similarity, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1311-1319, 2017.

[14] Deng J, Dong W, Socher R, Li L J, Li K and Fei-Fei L, Imagenet: A large-scale hierarchical image database, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 248-255, 2009.

[15] Krizhevsky A, Sutskever I and Hinton G E, Imagenet classification with deep convolutional neural networks, *In Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[16] He K, Zhang X, Ren S and Sun J, Deep residual learning for image recognition, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770-778, 2016.

[17] Osherov E and Lindenbaum M, Increasing CNN Robustness to Occlusions by Reducing Filter Support, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 550-561, 2017.

[18] Korman S, Reichman D, Tsur G and Avidan S, Fast-match: Fast affine template matching, *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2331-2338, 2013.

[19] Zhang C and Akashi I, Fast Affine Template Matching over Galois Field, *British Mach. Vis. Conf.*, pp. 121-1, 2015.

[20] Hel-Or Y, Hel-Or H and David E, Matching by tone mapping: Photometric invariant template matching, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 36, no. 2, pp. 317-330, 2014.
[21] Silpa-Anan C and HartleyR, Optimised KD-trees for fast image descriptor matching. *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1-8, 2008.
[22] Jegou H, Douze M and Schmid C, Product quantization for nearest neighbor search, *IEEE Trans. Pattern Anal. Mach*, vol. **33**, no. 1, pp. 117-128, 2011.
[23] Ge T, He K, Ke Q and Sun J, Optimized product quantization, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 36, no. 4, pp. 744-755, 2014.
[24] Nowozin S, Optimal decisions from probabilistic models: the intersection-over-union case. *IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 548-555, 2014.
[25] Elboher E and Werman M, Asymmetric correlation: a noise robust similarity measure for template matching, *IEEE Trans. Image Pro.*, vol. 22, no. 8, pp. 3062-3073, 2013.
[26] Pele O and Werman M, Robust real-time pattern matching using bayesian sequential hypothesis testing, *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. **30**, no. 8, pp. 1427-1443, 2008.