Robust Template Matching via Hierarchical Convolutional Features from a Shape Biased CNN

Bo Gao[1][0000−0003−3930−1815] and Michael W. Spratling[1][0000−0001−9531−2813]

Department of Informatics, King's College London, London, UK
{bo.gao,michael.spratling}@kcl.ac.uk

Abstract. Finding a template in a search image is an important task underlying many computer vision applications. Recent approaches perform template matching in a deep feature-space, produced by a convolutional neural network (CNN), which is found to provide more tolerance to changes in appearance. In this article we investigate if enhancing the CNN's encoding of shape information can produce more distinguishable features that improve the performance of template matching. This investigation results in a new template matching method that produces state-of-the-art results on a standard benchmark. To confirm these results we also create a new benchmark and show that the proposed method also outperforms existing techniques on this new dataset.

Keywords: Template match · Convolutional neural networks · VGG19.

1 Introduction

Template matching is a technique to find a rectangular region of an image that contains a certain object or image feature. It is widely used in many computer vision applications such as object tracking [1, 2], object detection [3, 4] and 3D reconstruction [5, 6]. A similarity map is typically used to quantify how well a template matches each location in an image. Traditional template matching methods calculate the similarity using a range of metrics such as the normalised cross-correlation (NCC), the sum of squared differences (SSD) or the zero-mean normalised cross correlation (ZNCC) applied to pixel intensity or color values. However, because these methods rely on comparing the values in the template with those at corresponding locations in the image patch they are sensitive to changes in lighting conditions, non-rigid deformations of the target object, or partial occlusions, which results in a low similarity score when one or multiple of these situations occur.

With the help of deep features learned from convolutional neural networks (CNNs), vision tasks such as image classification [7, 8], object recognition [9, 10], and object tracking [1, 2] have recently achieved great success. In order to succeed in such tasks, CNNs need to build internal representations that are less effected by changes in the appearance of objects in different images. To

improve the tolerance of template matching methods to changes in appearance recent methods have been successfully applied to a feature-space produced by the convolutional layers of a CNN [11–15].

The higher layers of CNNs are believed to learn representations of shapes from low-level features [16]. However, a recent study [17] demonstrated that ImageNet-trained CNNs are biased toward making categorisation decisions based on texture rather than shape. This work also showed that CNNs could be trained to increase sensitivity to shape and that this would improve accuracy and robustness both of object classification and detection. Assuming that shape information is also useful for template matching, these results suggest that the performance of template matching methods applied to CNN generated feature-spaces could potentially also be improved by training the CNN to be more sensitive to shape.

In this article we verified the assumption by comparing the features from four CNN models with a same network structure but differing in shape sensitivity. Our results show that training a CNN to learn about texture while biasing it to be more sensitive to shape information, can improve template matching performance. Furthermore, by comparing template matching performance when using feature-spaces created from all possible combinations of one, two and three convolutional layers of the CNN it was found that the best results were produced by combining features from both early and late layers. Early layers of a CNN encode lower-level information such as texture, while later layers encode more abstract information such as object identity. Hence, both sets of results (the need to train the CNN to be more sensitive to shape and the need to combine information for early and late layers) suggest that a combination of texture and shape information is beneficial for template matching.

Our main contributions are summarized as follows: (1) We created a new benchmark which, compared to the existing standard benchmark, is more challenging, provides a far larger number of images pairs, and is better able to discriminate the performance of different template matching methods. (2) By training a CNN to be more sensitive to shape information and combining features from both early and late layers, we created a feature-space in which the performance of most template matching algorithms is improved. (3) Using this feature-space together with an existing template matching method, DIM [18], we obtained state-of-art results on both the standard and new datasets.

2   Related Work

To overcome the limitations of classic template matching methods, many approaches [11, 12, 14, 15] have been developed. These methods can be classified into two main categories.

Matching. One category changes the computation that is performed to compare the template to the image to increase tolerance to changes in appearance. For example, Best-Buddies Similarity (BBS) counts the proportion of sub-regions in the template and the image patch that are Nearest-Neighbor (NN) matches [14]. Similarly, Deformable Diversity Similarity (DDIS) explicitly

considers possible template deformation and uses the diversity of NN feature matches between a template and a potential matching region in the search image [15]. The Divisive Input Modulation (DIM) algorithm [18] extracts additional templates from the background and lets the templates compete with each other to match the image. Specifically, this competition is implemented as a form of probabilistic inference known as explaining away [19,20] which causes each image element to only provide support for the template that is the most likely match. Previous work has demonstrated that DIM, when applied to color feature-space, is more accurate at identifying features in an image compared to both traditional and recent state-of-the-art matching methods [18].

Features. The second category of approaches changes the feature-space in which the comparison between the template and the image is performed. The aim is that this new feature-space allows template matching to be more discriminative while also increasing tolerance to appearance changes. Co-occurrence based template matching (CoTM) transforms the points in the image and template to a new feature-spaced defined by the co-occurrence statistics to quantify the dissimilarity between a template to an image [12]. Quality-aware template matching (QATM) is a method that uses a pretrained CNN model as a feature extractor. It learns a similarity score which reflects the (soft-) repeatness of a pattern using an algorithmic CNN layer [11].

Deep Features. Many template matching algorithms from the first category above, can be applied to deep features as well as directly to color images. The deep features used by BBS, CoTM and QATM are extracted from two specific layers of a pre-trained VGG19 CNN [21], *conv1_2* and *conv3_4*. Following the suggestion in [2] for object tracking, DDIS also takes features from a deeper layer: fusing features from layers *conv1_2*, *conv3_4* and *conv4_4*. [13] proposed a scale-adaptive strategy to select a particular individual layer of a VGG19 to use as the feature-space according to the size of template. In each case using deep features was found to significantly improve template matching performance compared to using color features.

A recent study showed that ImageNet-trained CNNs are strongly biased towards recognising textures rather than shapes [17]. This study also demonstrated that the same standard architecture (ResNet-50 [22]) that learns a texture-based representation on ImageNet is able to learn a shape-based representation when trained on 'Stylized-ImageNet': a version of ImageNet that replaces the texture in the original image with the style of a randomly selected painting through AdaIN style transfer [23]. This new shape-sensitive model was found to be more accurate and robust in both object classification and detection tasks. Inspired by the findings, in this paper we investigate if enhancing the shape sensitivity of a CNN can produce more distinguishable features that improve the performance of template matching.

Table 1. Four different VGG19 CNN models used in this paper. IN and SIN are the abbreviations of ImageNet and Stylized-ImageNet respectively.

Name	Training	Fine-tuning	Rank of shape sensitivity
Model_A	IN	-	4
Model_B	SIN	-	1
Model_C	IN+SIN	-	2
Model_D	IN+SIN	IN	3

3   Methods

Previous work on template matching in deep feature-space (see section 2) has employed a VGG19 CNN. To enable a fair comparison with those previous results, we also used the VGG19 architecture. However, we used four VGG19 models that differed in the way they were trained to encode different degrees of shape selectivity by the same approach used by [17] (as summarised in Table 1). Model_A was trained using the standard ImageNet dataset [21] (we used the pretrained VGG19 model from the PyTorch torchvision library) which has the least shape bias. Model_B was trained on the Stylized-ImageNet dataset and thus has the most shape bias. Model_C was trained on a dataset containing the images from both ImageNet and Stylized-ImageNet. Model_D was initialised with the weights of Model_C and then fine-tuning on ImageNet for 60 epochs using a learning rate of 0.001 multiplied by 0.1 after 30 epochs. Therefore Model_C and Model_D have intermediate levels of shape bias, with model_D being less selective to shape than Model_C. The learning rate was 0.01 multiplied by 0.1 after every 30 epochs for Model_B and after every 15 epochs for Model_C. Number of epochs was 90 for Model_B and 45 for Model_C (as the dataset used to train Model_C was twice as large as that used to train Model_B the number of weight updates was the same for both models). The other training hyperparameters used for each model were: batch size 256, momentum 0.9, and weight decay 1e-4. The optimizer was SGD.

In color feature-space the DIM algorithm was previously found to produce the best performance (see section 2). We therefore decided to use this algorithm to determine the best CNN feature-space to use for template matching. The DIM algorithm was applied to deep features in exactly the same way that it was previously applied to color images [18], except: (1) five (rather than four) additional templates were used; and, (2) the positive and rectified negative values produced by a layer of the CNN were directly separated into two parts and used as separate channels for the input to the DIM algorithm (in contrast, previously each channel of a color image was processed using a difference of Gaussians filter and the positive and rectified negative values produced were used as separate channels for the input to the DIM algorithm).
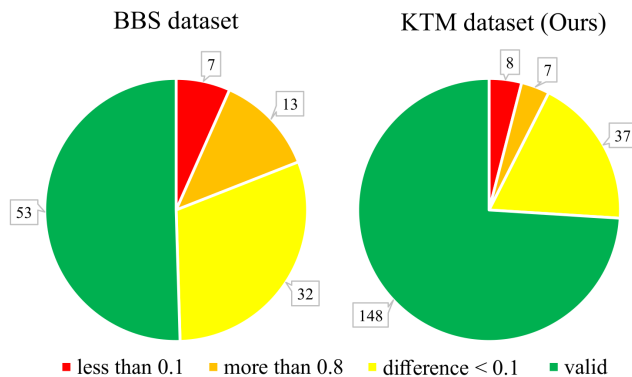
BBS dataset   KTM dataset (Ours)

■ less than 0.1   ■ more than 0.8   ■ difference < 0.1   ■ valid

Fig. 1. Discriminative ability of two datasets evaluated by comparing the IoU scores produced by ZNCC, BBS, DDIS and DIM.

4   Results

4.1   Dataset Preparation

The BBS dataset [14] has been extensively used for the quantitative evaluation of template matching algorithms [11–15]. This dataset contains 105 template-image pairs which are sampled from 35 videos (3 pairs per video) from a tracking dataset [24]. Each template-image pair are taken from frames of the video that are 20 frames apart. To evaluate the performance of a template matching algorithm the intersection-over-union (IoU) is calculated between the predicted bounding box and the ground truth box for the second image in the pair. The overall accuracy is then determined by calculating the area under the curve (AUC) of a success curve produced by varying the threshold of IoU that counts as success.

Although the BBS data is widely used, it is not particularly good at discriminating the performance of different template matching methods. To illustrate this issue we applied one baseline method (ZNCC) and three state-of-art methods (BBS, DDIS and DIM) to the BBS dataset in color space. The results show that there are 52 template-image pairs where all methods generate very similar results: these can be sub-divided into 7 template-image pairs for which all methods fail to match (IoU less than 0.1 for all four methods), 13 template-image pairs for which all methods succeed (IoU greater than 0.8 for all four methods), and 32 template-image pairs for which all methods produce similar, intermediate, IoU values within 0.1 of each other. This means that only 53 template-image pairs in the BBS dataset help to discriminate the performance of these four template matching methods. These results are summarised in Figure 1.

We therefore created a new dataset, the King's Template Matching (KTM) dataset, following a similar procedure to that used to generate the BBS dataset. The new dataset contains 200 template-image pairs sampled from 40 new videos
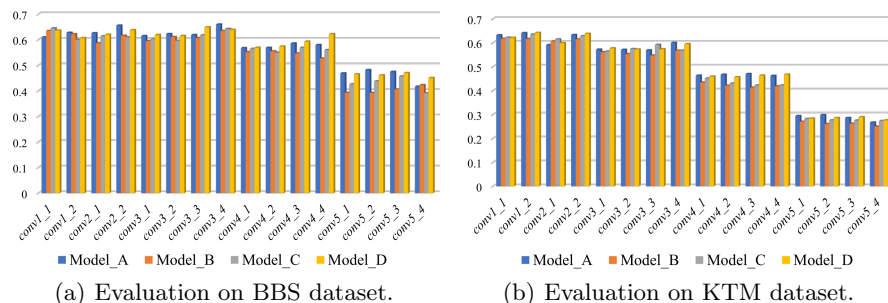
(a) Evaluation on BBS dataset.   (b) Evaluation on KTM dataset.

Fig. 2. The AUC scores of DIM using features from different convolutional layers of four models.

(5 pairs per video) selected from a different tracking dataset [25]. In contrast to the BBS dataset, the template and the image were chosen manually to avoid pairs that contain significant occlusions and non-rigid deformations of the target that no method is likely to match successfully, and the image pairs were separated by 30 (rather than 20) frames to reduce the number of pairs for which matching would be easy for all methods. These changes make the new data more challenging and provide a far larger number of images pairs that can discriminate the performance of different methods, as shown in Figure 1. Both the new dataset and the BBS dataset were used in the following experiments.

4.2 Template matching using features from individual convolutional layers

To reveal how the shape bias effects template matching, we calculate AUC using DIM with features from every single convolutional layer of the four models. As the features from the later convolutional layers are down-sampled using max-pooling, by a factor of $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ compared to the original image, the bounding box of the template is also multiplied by the same scaling factor and the resulting similarity map is resized back to the original image size to make the prediction. The AUC scores across the BBS and DIM datasets are summarised in Figure 2.

For all four models there is a tendency for the AUC to be higher when template matching is performed using lower layers of the CNN compared to later layers. This suggests that template matching relies more on low-level visual attributes, such as texture, rather than higher-level ones such as shape. Among the three models trained with Stylized-ImageNet, the AUC score for most CNN layers is greater for Model_D than Model_C, and greater for Model_C than Model_B. This also suggests that template matching relies more on texture features than shape features. Comparing Model_A and Model_D, it is hard to say which one is better. However, the AUC score calculated on the BBS dataset using features from *conv4_4* of Model_D is noticeably better than that for Model_A. This suggests that increasing the shape-bias of later layers of the CNN could potentially

Table 2. Partial AUC scores of DIM using features from two convolutional layers of Model_A (upper value in each cell) and Model_D (lower value in each cell).

(a) Evaluation on BBS dataset.

AUC \ Layer Layer	conv3_3	conv3_4	conv4_1	conv4_2	conv4_3	conv4_4
conv1_1	0.710 0.707↓	0.705 0.714↑	0.713 0.704↓	0.697 **0.718**↑	0.698 0.710↑	0.711 0.708↓
conv1_2	0.686 0.686	0.686 0.687↑	0.674 0.707↑	0.655 0.696↑	0.680 0.690↑	0.683 0.710↑
conv2_1	0.658 0.659↑	0.670 0.669↓	0.664 0.665↑	0.653 0.671↑	0.662 0.683↑	0.667 0.693↑
conv2_2	0.659 0.665↑	0.661 0.667↑	0.653 0.676↑	0.641 0.679↑	0.659 0.676↑	0.663 0.682↑

(b) Evaluation on KTM dataset.

AUC \ Layer Layer	conv3_3	conv3_4	conv4_1	conv4_2	conv4_3	conv4_4
conv1_1	0.687 0.689↑	0.684 0.691↑	0.677 0.682↑	0.668 0.695↑	0.670 0.684↑	0.678 0.687↑
conv1_2	0.687 0.680↓	0.689 0.694↑	0.682 0.695↑	0.685 **0.697**↑	0.675 0.691↑	0.682 0.690↑
conv2_1	0.634 0.642↑	0.633 0.651↑	0.647 0.665↑	0.645 0.671↑	0.655 0.668↑	0.639 0.666↑
conv2_2	0.642 0.657↑	0.651 0.664↑	0.664 0.670↑	0.661 0.673↑	0.669 0.669	0.669 0.669

lead to better template matching. However, this results is not reflected by the results for the KTM dataset. One possible explanation is that the templates in the KTM dataset are smaller, in general, than those in the BBS dataset (if the template size is defined as the product of its width and height, then the mean template size of for the DIM dataset is 1603 whereas it is 3442 for the BBS dataset). Smaller templates tend to be less discriminative. The sub-sampling that occurs in later levels of the CNN results in templates that are even smaller and less disriminative. This may account for the worse performance of the later layers of each CNN when tested using the KTM dataset rather than the BBS dataset. It is also a confounding factor in attributing the better performance of the early layers to a reliance on texture information.

4.3 Template matching using features from multiple convolutional layers

We compared Model_A and Model_D by applying the DIM template matching algorithm to features extracted from multiple convolutional layers of each CNN. To combine feature maps with different sizes bilinear interpolation was used to make them the same size. If the template was small (height times width less than 4000) the feature maps from the later layer(s) were scaled to be the same size as those in the earlier layer(s). If the template was large, the feature maps from the earlier layer(s) were reduced in size to be the same size as those in the later layer(s). To maintain a balance between low and high level features, the dimension of the features maps form the latter layer(s) were reduced by PCA to the same number as those in the earlier layer.

Table 2 shows the AUC scores produced by DIM using features from two convolutional layers of Model_A and Model_D. All possible combinations of two layers were tested and the table shows only selected results with the best performance. Each cell of the table contains two AUC scores, the upper one is produced using Model_A and the bottom is produced by Model_D. The up and down-arrows indicate whether the AUC score of Model_D is better or worse than that of Model_A.

It can be seen from Table 2 that for the 24 layer combinations for which results are shown, 21 results for both BBS and KTM dataset are better for

Table 3. Best 10 results when using combinations of features from three convolutional layers of Model_D. $C12^{41}_{44}$ means fusing the features from *conv1_2*, *conv4_1* and *conv4_4* for instance.

(a) Evaluation on BBS dataset.

Layers	$C12^{41}_{44}$	$C11^{34}_{43}$	$C11^{42}_{44}$	$C12^{43}_{52}$	$C11^{22}_{43}$	$C11^{41}_{44}$	$C11^{41}_{43}$	$C11^{34}_{42}$	$C12^{43}_{44}$	$C11^{34}_{44}$
AUC	0.728	0.727	0.724	0.724	0.723	0.722	0.720	0.720	0.720	0.720

(b) Evaluation on KTM dataset.

Layers	$C11^{34}_{42}$	$C12^{32}_{43}$	$C11^{34}_{43}$	$C12^{22}_{42}$	$C12^{31}_{42}$	$C12^{34}_{42}$	$C12^{34}_{43}$	$C12^{31}_{43}$	$C12^{33}_{43}$	$C11^{33}_{42}$
AUC	0.711	0.709	0.708	0.706	0.706	0.705	0.705	0.705	0.705	0.704

Model_D than for Model_A. Furthermore, the best result for each dataset (indicated in bold) is generated using the features from Model_D. These results thus support the conclusion that more discriminative features can be obtained by slightly increasing the shape bias of the VGG19 model which increases the performance of template matching.

To determine if fusing features from more layers would further improve template matching performance, DIM was applied to all combinations of three layers from Model_D. There are a total of 560 different combinations of three layers. It is impossible to show all these results in this paper, therefore the highest 10 AUC scores are shown in Table 3. For both dataset, using three layers produced an improvement in the best AUC score (around 0.01) compared to when using two layers.

4.4   Comparison with other methods

This section compares our results with those produced by other template matching methods in both color and deep feature-space. When evaluated on the BBS dataset, the deep features used by each template matching algorithm were the features from layers *conv1_2*, *conv4_1*, and *conv4_4* of Model_D. When evaluated on the KTM dataset the deep features used as the input to each algorithm were those from layers *conv1_1*, *conv3_4* and *conv4_2* of Model_D. BBS, CoTM and QATM have been tested on BBS data by their authors using different deep features, so we also compare our results to these earlier published results.

The comparison results are summarised in Table 4. All methods expect QATM and BBS produce improved results using the proposed deep features than when using color features. This is true for both datasets. Of the methods that have previously been applied to deep features the performances of two (NCC and QATM) are improved, and that of two others (BBS and CoTM) are made worse by using our method of defining the deep feature-space. Potentially, further improvements to the performance of these methods could be achieved by optimising the feature extraction method for the individual template matching algorithm, as has been done here for DIM. However, it should be noted that simple metrics for comparing image patches such as NCC and ZNCC produce near state-of-the-art performance when applied to our proposed deep feature-space,

outperforming much more complex methods of template matching such as BBS, CoTM, and QATM when these methods are applied to any of the tested feature-spaces, including that proposed by the authors of these algorithms. One known weakness of BBS is that it may fail when the template is very small compared to target image [14]. This may explain the particularly poor results of this method when applied to the KTM dataset.

DIM achieves the best results on both datasets when applied to deep features. DIM performs particularly well on the BBS dataset producing an AUC of 0.73 which, as far as we are aware, makes it the only method to have scored more than 0.7 on this dataset. The DIM algorithm also produces state-of-the-art performance on the KTM dataset when applied to deep features. When applied to color features, the results are good, although not as good as DDIS on the KTM dataset. This is because small templates in the KTM dataset sometime contain insufficient detail for the DIM algorithm to successfully distinguish the object. Using deep features enhances the discriminability of small templates sufficiently that the performance of DIM increases significantly. The results demonstrate that the proposed approach is effective at extracting distinguishable features which lead to robust and accurate template matching.

Table 4. Quantitative comparison of the performance of different template matching algorithms using different input features.

AUC \ Feature / Method	BBS dataset			KTM dataset	
	Color	Deep	Deep (Proposed)	Color	Deep (Proposed)
SSD	0.46	-	0.54	0.42	0.54
NCC	0.48	0.63 [13]	0.67	0.42	0.67
ZNCC	0.54	-	0.67	0.48	0.67
BBS	0.55	0.60 [14]	0.54	0.44	0.55
CoTM	0.54^1	0.67 [12]	0.64	0.51	0.56
DDIS	0.64	-	0.66	**0.63**	0.68
QATM	-	0.62^2	0.66	-	0.64
DIM	**0.69**	-	**0.73**	0.60	**0.71**

---

[1] We were unable to reproduce this result using code supplied by the authors of CoTM. Our different result is shown in the table.

[2] The authors of QATM report an AUC score of 0.69 when this method is applied to the BBS dataset [11]. However, examining their source code we note that this result is produced by setting the size of the predicted bounding box equal in size to the width and height of the ground truth bounding box. Other methods are evaluated by setting the size of the predicted bounding box equal equal to the size of the template (i.e. without using knowledge of the ground truth that the algorithm is attempting to predict). We have re-tested QATM using the standard evaluation protocol and our result for the original version of QATM is 0.62. As QATM is designed to work specifically with a CNN it was not applied directly to color images.

5   Conclusions

Our results demonstrate that slightly increasing the shape bias of a CNN (by changing the method of training the network) produces more distinguishable features in which template matching can be achieved with greater accuracy. By running a large number of experiments we determined the best combination of convolutional features from our shape-biased VGG19 on which to perform template matching with the DIM algorithm. This same feature-space was shown to improve the performance of most other template matching algorithms as well. The DIM algorithm applied to our new feature-space produces state-of-art results on two benchmark datasets.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)
2. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Robust visual tracking via hierarchical convolutional features. IEEE transactions on pattern analysis and machine intelligence (2018)
3. Ahuja, K., Tuli, P.: Object recognition by template matching using correlations and phase angle method. International Journal of Advanced Research in Computer and Communication Engineering 2(3), 1368–1373 (2013)
4. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016)
5. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision 47(1-3), 7–42 (2002)
6. Chhatkuli, A., Pizarro, D., Bartoli, A.: Stable template-based isometric 3d reconstruction in all imaging conditions by linear least-squares. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 708–715 (2014)
7. Chan, T.H., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y.: PCANet: A simple deep learning baseline for image classification? IEEE transactions on image processing 24(12), 5017–5032 (2015)
8. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3156–3164 (2017)

9. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3367–3375 (2015)

10. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3d pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3109–3118 (2015)

11. Cheng, J., Wu, Y., AbdAlmageed, W., Natarajan, P.: QATM: Quality-aware template matching for deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11553–11562 (2019)

12. Kat, R., Jevnisek, R., Avidan, S.: Matching pixels using co-occurrence statistics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1751–1759 (2018)

13. Kim, J., Kim, J., Choi, S., Hasan, M.A., Kim, C.: Robust template matching using scale-adaptive deep convolutional features. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). pp. 708–711. IEEE (2017)

14. Oron, S., Dekel, T., Xue, T., Freeman, W.T., Avidan, S.: Best-buddies similarity—robust template matching using mutual nearest neighbors. IEEE transactions on pattern analysis and machine intelligence 40(8), 1799–1813 (2017)

15. Talmi, I., Mechrez, R., Zelnik-Manor, L.: Template matching with deformable diversity similarity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 175–183 (2017)

16. Kriegeskorte, N.: Deep neural networks: a new framework for modeling biological vision and brain information processing. Annual review of vision science 1, 417–446 (2015)

17. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)

18. Spratling, M.W.: Explaining away results in accurate and tolerant template matching. Pattern Recognition p. 107337 (2020)

19. Kersten, D., Mamassian, P., Yuille, A.: Object perception as bayesian inference. Annu. Rev. Psychol. 55, 271–304 (2004)

20. Spratling, M.W.: Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. Neural computation 24(1), 60–103 (2012)

21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

23. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)

24. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1834–1848 (2015)

25. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: Algorithms and benchmark. IEEE Transactions on Image Processing 24(12), 5630–5644 (2015)