Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220015

A Robust Defect Detection Method with Dense Differential Siamese Network

Juntao He[a], Xiaodong Wang[a,1], Zhiqiang Zeng[a] and Fei Yan[a]
[a] *College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China*

Abstract. In the context of defect detection, there is a difficulty in collecting annotated datasets, which leads to limited labeled data. In addition to this, most of the defect detection methods have the problem of missing detailed information about the defects. To cope with these problems, this paper shows a dense differential Siamese network structure for the defect detection of stamping manufacture. In the stamping setting, the foreground of the image frequently changes, while the background remains the same. Based on this finding, we separate the encoding layer of the network into two streams with the same structure and shared weights, so that we can handle the foreground and background image pairs simultaneously. To extract detailed information of defects, we also impose the dense skip connections into our network. Through these skip connections, we can obtain different levels of semantic information and capture more detailed information about the defects. Testing results on the defect dataset collected from real stamping machines show that our method significantly improves over other state-of-the-art methods on several evaluation metrics.

Keywords. defect detection, fully convolutional Siamese network, dense skip connections

1. Introduction

Defect detection is an extremely important part of modern industrial production, which improves production efficiency and yield by detecting defective products in time during production [1]. Research on defect detection has a long history, and it continues to evolve with the development of computer vision [3, 6]. The goal of a defect detection system is to detect the part of the product in the image that appears defects. Today, defect detection faces many difficulties. For example, in the defect detection field, there are few labeled datasets, which prevents many algorithms from learning well. In addition, the diversity of defect types exacerbates this problem. On the other hand, many algorithms consider the presence or absence of defects and are not concerned with the complete segmentation of defects. To reduce the impact due to the limited labeled data, a possible approach is to use Siamese network in the feature extraction stage [7, 8]. Unlike the traditional defect algorithm that requires only a single image as the input, the Siamese network requires two paired input images. Looking at defect detection from the perspective of change, we can consider the defect as the part that changes relative to the normal case. For example, a scratch that appears on the surface of a product is the part

[1] Corresponding Author: Xiaodong Wang, E-mail: xdwangjsj@xmut.edu.cn

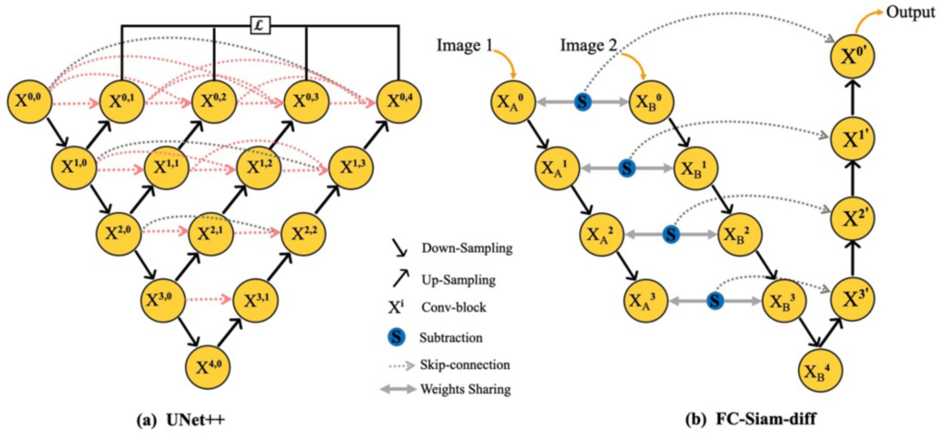

Figure 1. Two close related networks of our method. (a) is a network applied in the field of medical image segmentation; (b) is the network applied in the field of change detection.

that changes relative to the normal surface. In this case, the Siamese networks could collect the rich information from both foreground (defects) and background (non-defects) and may achieve more accurate detection results than traditional methods. In real-world manufacturing products, small defects are often encountered. However, most of recent defect detection algorithms usually focus on detecting large defects and are not very effective for detecting small defects.

In this paper we propose a fully convolutional neural network for defect detection with a Siamese structured encoder. In addition, inspired by UNet++, we add dense skip connections for collecting semantic information at different levels. Our approach uses a pair of images from a defect detection dataset for end-to-end training and directly outputs a binary map labeled with defects. Moreover, we compare our proposed method with current state-of-the-art change detection methods on dataset generated from actual factory production, achieving superior performance.

This paper is organized as follows. Section 2 explores the sources of our inspiration in conceiving this approach and provides some related work. Section 3 describes in detail the neural network we constructed for defect detection. Section 4 evaluates the performance of our method and compares it with other state-of-the-art methods.

2. Related Work

Today's defect detection is mainly performed using common image processing algorithms, such as extraction of specific defects based on threshold segmentation, boundary segmentation, and defect extraction by geometric features and histogram features of the image [2]. The generalizability of these methods is low, and their accuracy is not very high. Siamese structured auto-encoders are widely used in the field of change detection [8-12], and a variety of architectures have been derived according to different application scenarios and practical situations. For example, the Fully Convolutional

Siamese difference~(FC-Siam-diff) network [7], shown in Figure 1(b), is one of the most widely used techniques with Siamese structures and performs well in many change detection scenarios. In the change detection, to detect the difference between two images, two encoders with sharing weights are used in the feature extraction phase to extract the features of both images separately, and skip connections are used to better extract the features in the region where the change occurs.

UNet++ is novel and effective in the field of medical segmentation [13]. As shown in Figure 1(a), the UNet++ architecture is essentially a deeply supervised encoder-decoder network, where the encoder and decoder sub-networks are connected by a series of nested, dense skip connections. UNet++ has a great power to extract multi-scale features from different convolution levels. The biggest difference between UNet and UNet++ is the re-designed skip connections in each level of the decoder. Therefore, UNet++ can collect the information extracted by the shallow sub-encoder. Second paragraph.

In defect detection tasks, the defects are often relatively small, so it is difficult to segment them accurately. Recent algorithms in change detection such as Early Fusion [11], Fully Convolutional Early Fusion, and Fully Convolutional Siamese Concatenation [7], etc., are applied in remote sensing and image processing. In these applications, their image resolution is high, and the defect areas are often large. Hence, directly applying these algorithms to defect detection will result in low detection accuracy and require further restructuring to suit the defect detection task.

3. Proposed Approach

The proposed network is based on the fully convolution Siamese network structure [7] and UNet++, which is shown in Figure 2(a). The $X^{i,j}$ node in the figure denotes the convolution block, where i indexes the down-sampling layer along the encoder and j indexes the convolution layer of the dense block along the skip pathway [13], and in the following, we use $x^{i,j}$ to denote the output of the $X^{i,j}$node. Our goal is to find the defects in the input images. As we discussed in Section 1, we need to input two images to the Siamese encoder, image 1 and image 2. Image 1 is the background image of this product without defects, and image 2 is the image to be detected where we need to determine if there are defects. In real detection procedure, the background (image 1) is usually fixed, while foreground (image 2) may contain a variety of defects, and these defect areas are changing relative to the corresponding area in image 1. In other words, the changing area is the area where the defects appear and is the main concern of our network. In order to better collect the characteristics of these changes, we use the operation of subtracting the feature maps of the same layer in the encoder part. That is, in the encoder part, we subtract the feature maps at the same level of these two encoders and then transfer subtraction results to the decoder part. Formally, the feature maps obtained from image 1 and image 2 at i-th level is defined as $x_A^{i,0}$ and $x_B^{i,0}$, respectively. Then, the result of feature map subtraction can be defined as:

$$s_i = \left| x_A^{i,0} - x_B^{i,0} \right| \tag{1}$$

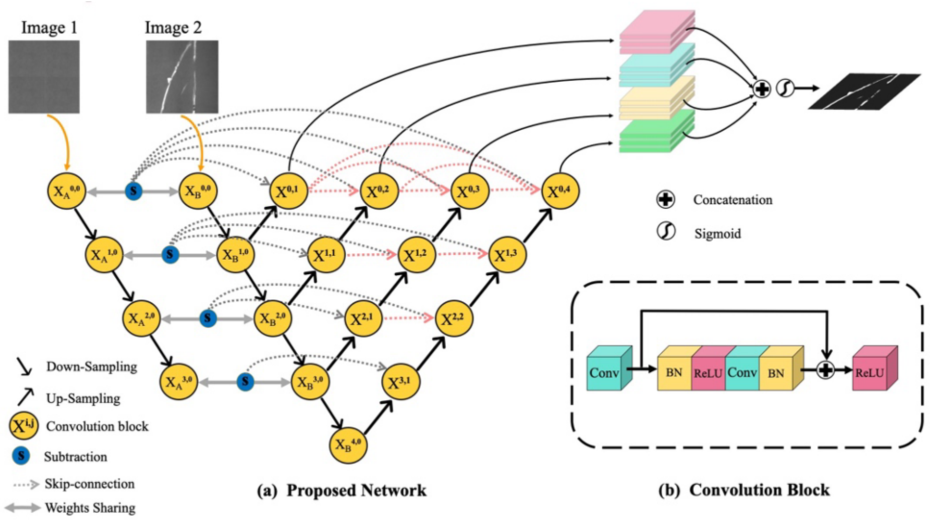

Figure 2. Illustration of the proposed architecture. (a) is the backbone of our network, downward arrows and upward arrows indicate down-sampling and up-sampling, respectively. The gray and red dotted arrows represent skip connections. The node $X^{i,j}$ indicates a convolution block, of which the detailed structure is shown in (b).

To preserve fine-grained localization features and prevent the loss of target object information, we use dense skip connections mechanism between the encoder and decoder, which inspired by UNet++ [13]. Thanks to the dense skip connections, the position information in the shallow layer is directly applied to the deep layer so that the fine-grained information in the input image can be maintained. Take the node $X^{0,3}$ as an example, it directly receives not only the feature maps up-sampled by its upper level, *i.e.*, $X^{1,2}$, but also the information output by $X^{0,1}$ and $X^{0,2}$. More importantly, we concatenate the feature map s_0, which is the feature map obtained by subtracting $x_A^{0,0}$ and $x_B^{0,0}$, into the $X^{0,3}$ node as well. Through these dense skip connections, the differentiated features and positional information we obtained through the subtraction operation in the encoder stage can be better preserved.

In our network, we design $X^{i,j}$ as a residual block [14], which facilitates better convergence capacities for the deep network architecture. In particular, the shortcut connection is located after the first convolutional layer so as to maintain the unity of the convolutional block, which is shown in Figure 2(b). Let $x^{i,j}$ denote the output of node $X^{i,j}$, $x^{i,j}$ can be formulated as follows:

$$x^{i,j} = \begin{cases} \mathcal{D}\left(\mathcal{C}(x^{i-1,0})\right) & j = 0 \\ \mathcal{C}\left(\left[\mathcal{S}\left(x_A^{i,0},\ x_B^{i,0}\right), \mathcal{U}(x^{i+1,j-1})\right]\right) & j = 1 \\ \mathcal{C}\left(\left[\mathcal{S}\left(x_A^{i,0},\ x_B^{i,0}\right), \mathcal{U}(x^{i+1,j-1}), [x^{i,k}]_{k=1}^{j-1}\right]\right) & j > 1 \end{cases} \tag{2}$$

where the function $\mathcal{D}(\cdot)$ is used to down-sample the feature map, using a 2×2 size max-pooling operation. $\mathcal{C}(\cdot)$ denotes our residual function in the convolution block in Figure 2(b). $\mathcal{S}(\cdot)$ is the element-wise subtraction operation we mentioned in Eq.(1). The

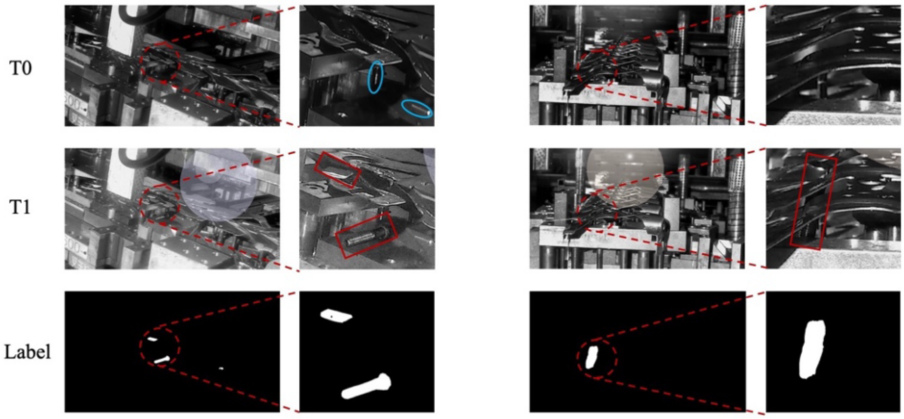

Figure 3. Bi-temporal images with defect variations in stamping defect dataset.

function $\mathcal{U}(\cdot)$ denotes the up-sampling operation. Finally, the $[\cdot]$ denotes the concatenation of the feature maps. At the end of this network, we generate four full resolution feature maps, *i.e.*, $\left\{x^{0,j}, j \in \{1,2,3,4\}\right\}$, which is easy to perform deep supervision and allows our model to integrate multiple levels of semantic information. It should be noted that the shallow sub-decoder outputs a more fine-grained feature map, while the deep sub-decoder outputs a more coarse-grained feature map. The final feature maps with the different semantic information are joined together and processed by a sigmoid layer to generate the final binary result map.

4. Experiments

4.1. Dataset and Evaluation Metrics

The stamping defect dataset was collected from a factory engaged in mold production. The subject of the dataset is a variety of stamping machines in the factory. These stamping machines are used for producing molds. Figure 3 shows the bi-temporal images, *i.e.*, images captured at two different times, in the stamping defect dataset. Here, we use T0 and T1 to represent the images acquired at different times, where the red boxes represent the defects that need to be identified. In the actual production environment, in addition to the different types of stamping machines and defects, there will be interference factors, such as noise that is similar to the defects, *i.e.*, the objects marked with blue ellipses in Figure 3, and light changes.

In this dataset, there are about 400 pairs of images in total, where each image is 1920×1080 pixels in size. Since the resolution of the images is relatively large, training the model directly with the original images can lead to a high computation cost. To solve this issue, we use the method in [15] to generate image patches by cropping a 160×160 sized sub-picture on top of the original image. Totally, we obtain a training set with 30,000 samples, a testing set with 9,000 samples, and validation set with 4500 samples.

We use Pytorch to implement our model and the compared methods on the hardware platform with the ubuntu operation system and a NVIDIA GTX TITAN X graphics card.

Table 1. Comparison results on stamping defect dataset.

Methods	**Precision**	**Recall**	**F$_1$-Score**	**OA**
EF	0.596	0.664	0.628	0.905
FC-EF	0.577	0.709	0.636	0.885
FC-Siam-conc	0.649	0.693	0.670	0.916
FC-Siam-diff	0.709	0.685	0.697	0.927
UNet++_MSOF	0.772	0.688	0.727	0.936
Ours	**0.807**	**0.714**	**0.757**	**0.959**

During training, we use the minibatch ADAM algorithm as an optimizer with the batch size 64, the learning rate 1×10^{-4}. We use KaiMing normalization to initialize the neural network and train the network with 50 epochs. In terms of the loss function, the cross-entropy loss function is used in our experiments as well as the comparison experiments.

Four closely related UNet [16] type methods, *i.e.*, Early Fusion (EF) [11], Fully Convolutional Early Fusion (FC-EF) [7], Fully Convolutional Siamese Concatenation (FC-Siam-conc) [7], and Fully Convolutional Siamese Difference (FC-Siam-diff) [7], are selected for comparison in this paper. Both EF and FC-EF networks first concatenate two image pairs and then input them into the encoder, so they have only one encoder. FC-Siam-conc and FC-Siam-diff use Siamese structured encoders, that is, two image pairs are input to two encoders separately, the difference is that in each layer of the encoder, the former concatenates the features, while the latter uses subtraction operation for the features. These four methods are based on UNet [16] and Siamese networks [17] and have been experimentally shown high performance in the field of defect detection. Besides, another UNet++ [13] type method, *i.e.*, UNet++ with Multiple Side-Output Fusion (UNet++_MSOF) [12], is also introduced and compared. This network uses the strategy of concatenating images first as mentioned in EF, FC-EF in the encoder part, while in the decoder part it uses the dense skip connections used in UNet++ and captures fine features by fusing the side-outputs. The comparison results on the stamping defect dataset are shown in Table 1. In this experiment, we use four quantitative metrics, *i.e.*, Precision, Recall, F$_1$-Score, Overall Accuracy (OA), for evaluation. These four evaluation metrics are descried as:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 - Score = \frac{2 \times Precison \times Recall}{Precision + Recall} \tag{5}$$

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TP, FP, TN, and FN denote the number of true positives, the number of false positives, the number of true negatives, and the number of false negatives, respectively.

4.2. Results Comparisons

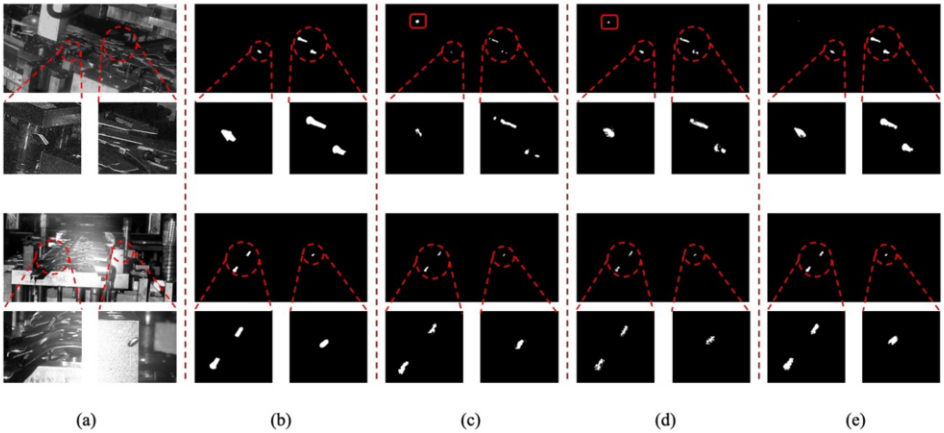

Figure 4. Visualization results on the stamping defect dataset. Smaller images are enlargements of the corresponding areas: (a) is the original images; (b) is the ground truth; (c) is the result of FC-EF; (d) is the result of FC-Siam-diff; (e) is the result of our method.

As can be seen in Table 1, our method is able to detect defects effectively, with a significant improvement over the other methods, and achieves the best results in all metrics. In order to verify the effectiveness and superiority of our proposed method, two typical scenarios in real production are presented for visual comparisons. As shown in Figure 4, it can be observed that our method is more accurate in segmenting the boundaries of defects. Overall, our network is better at preserving the detailed information of the defects and can segment the defects better. Besides, our method is more robust to light changes (red square area).

The results show that the EF and FC-EF have the lowest F_1-Score and OA values among the six methods. The reason is that both of these networks are composed of only the encoding layer used for contraction and the decoding layer used for expansion, and do not use the Siamese structure. Therefore, the features of defects are not well learned for the case where only a small number of samples with labels are available. As shown in Figure 4(c), in these two different scenarios, FC-EF identifies very limited defects and misidentifies light changes as defects (red square area).

Both FC-Siam-conc and FC-Siam-diff are networks that use the Siamese structure at the encoding layer, and thus have higher F_1-Score and OA values than EF and FC-EF with limited labeled data. FC-Siam-diff uses the subtraction strategy mentioned in Eq.1 at the encoding layer. Besides, by using differential information collection in the feature extraction layer, it obtains better collection of defects feature, *i.e.*, FC-Siam-diff is 2.7% and 1.1% higher than FC-Siam-conc in terms of F_1-Score and OA values, respectively. With the help of subtraction strategy in the encoding layer, our method improves the UNet++_MSOF, by 3.5%, 2.6%, 3.0%, and 2.3% in Precision, Recall, F_1-Score, and OA values, respectively.

It should be noticed that both our network and $\mathrm{UNet}++_{\mathrm{MSOF}}$ use the dense skip connections strategy. Nevertheless, FC-Siam-conc and FC-Siam-diff do not have complex connections between the encoding and decoding layers, so that low-level detail information is missed during the encoding process, as shown in the Figure 4(d). This observation also demonstrates that the fine localization information in the encoding layer can be successfully transmitted to the decoding layer through skip connections, thereby avoiding the loss of fine-grained features and better detecting small defects or detailed regions of defects.

5. Conclusion

In this paper, we propose an algorithm for defect detection applications in the industry manufacturing. To better adapt the algorithm to defect detection, we combine the Siamese network structure with the dense skip connections. To facilitate the convergence of gradients in deep full convolutional networks, we use a residual block strategy, which can also benefit the network to obtain more detailed information of defects. Compared with the other state-of-the-art methods, our proposed approach performs better in both visualization and quantitative metrics evaluation.

Acknowledgement

This paper was supported by National Natural Science Foundation of China (Grant Nos.61871464, U1805264), National Natural Science Foundation of Fujian Province (Grant Nos.2020J01266, 2021J011186), the "Climbing" Program of XMUT (Grant No. XPDKT20031), Scientific Research Fund of Fujian Provincial Education Department (Grant No. JAT200486), Program of XMUT for high-Level talents introduction plan (Grant No. YKJ19003R).

References

[1] Yang J, Li S, Wang Z, et al. Real-time tiny part defect detection system in manufacturing using deep learning[J]. IEEE Access, 2019, 7: 89278-89291.

[2] Ghorai S, Mukherjee A, Gangadaran M, et al. Automatic defect detection on hot-rolled flat steel products[J]. IEEE Transactions on Instrumentation and Measurement, 2012, 62(3): 612-621.

[3] Koch C, Georgieva K, Kasireddy V, et al. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure[J]. Advanced Engineering Informatics, 2015, 29(2): 196-210.

[4] Mahajan P M, Kolhe S R, Patil P M. A review of automatic fabric defect detection techniques[J]. Advances in Computational Research, 2009, 1(2): 18-29.

[5] Bo T, Jianyi K, Shiqian W. Review of surface defect detection based on machine vision[J]. Journal of Image and Graphics, 2017, 22(12): 1640-1663.

[6] Kumar A. Computer-vision-based fabric defect detection: A survey[J]. IEEE transactions on industrial electronics, 2008, 55(1): 348-363.

[7] Daudt R C, Le Saux B, Boulch A. Fully convolutional siamese networks for change detection[C]//2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 4063-4067.

[8] Fang S, Li K, Shao J, et al. SNUNet-CD: A Densely Connected Siamese Network for Change Detection of VHR Images[J]. IEEE Geoscience and Remote Sensing Letters, 2021.

[9] Chen J, Yuan Z, Peng J, et al. DASNet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 14: 1194-1206.

[10] Khelifi L, Mignotte M. Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis[J]. IEEE Access, 2020, 8: 126385-126400.

[11] Daudt R C, Le Saux B, Boulch A, et al. Urban change detection for multispectral earth observation using convolutional neural networks[C]//IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2018: 2115-2118.

[12] Peng D, Zhang Y, Guan H. End-to-end change detection for high resolution satellite images using improved UNet++[J]. Remote Sensing, 2019, 11(11): 1382.

[13] Zhou Z, Siddiquee M M R, Tajbakhsh N, et al. Unet++: A nested u-net architecture for medical image segmentation[M]//Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer, Cham, 2018: 3-11.

[14] Yu X, Yu Z, Ramalingam S. Learning strict identity mappings in deep residual networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4432-4440.

[15] Lebedev M A, Vizilter Y V, Vygolov O V, et al. CHANGE DETECTION IN REMOTE SENSING IMAGES USING CONDITIONAL ADVERSARIAL NETWORKS[J]. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 2018, 42(2).

[16] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.

[17] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, 1: 539-546.