ISSN 1060-992X, Optical Memory and Neural Networks, 2019, Vol. 28, No. 2, pp. 101–108. © Allerton Press, Inc., 2019.

Comparison of Face Recognition and Detection Models: Using Different Convolution Neural Networks

Kai Kang*

College of Information Science and Technology and PRT Advanced Printing Technology Innovation Laboratory, Xiamen University Tan Kah Kee College, Zhangzhou, Fujian, 363105 China

**e-mail: kkangxm@aliyun.com*

Received December 20, 2018; revised March 5, 2019; accepted March 11, 2019

Abstract—Face detection and recognition plays an important role in many occasions. This study explored the application of convolutional neural network in face detection and recognition. Firstly, convolutional neural network was briefly analyzed, and then a face detection model including three convolution layers, four pooling layers, introduction layers and three fully connected layers was designed. In face recognition, the self-learning convolutional neural network (CNN) model for global and local extended learning and Spatial Pyramid Pooling (SPP)-NET model were established. LFW data sets were used as model test samples. The results showed that the face detection model had an accuracy rate of 99%. In face recognition, the self-learning CNN model had an accuracy rate of 94.9% accuracy, and the SPP-Net model had an accuracy rate of 92.85%. It suggests that the face detection and recognition model based on convolutional neural network has good accuracy, and the face recognition efficiency of self-learning CNN model was better, which deserves further research and promotion.

Keywords: face detection, face recognition, convolutional neural network

DOI: 10.3103/S1060992X19020036

1. INTRODUCTION

Face detection and recognition has great practical value [1] and has been extensively applied in aspects such as security and business [2]. It is always a hot research field [3]. Under the promotion of advanced technology, face detection and recognition technology has also been greatly developed and more and more technologies have been continuously studied and practiced in that field, such as extreme learning machine [4], subspace learning [5] and support vector machine [6]. Face recognition which is difficult to be realized because of the influence of pose, illumination and occlusion [7] has attracted much attention. Lai et al. [8] designed Sparse Representation Based Classification (SRC) algorithm using Lagrange Duality Method (LDM) and put forward LDM-SRC face recognition method. They found that the method could not only reduce the efficiency of SRC algorithm and shorten computing time, but also has good face recognition accuracy. Zhang et al. [9] designed a kernel sparse representation based classifier ensemble (KSRCE) which had good classification performance on face image data without considering the impact of random projection and kernel Gram matrix on KSRC. Lei et al. [10] proposed a face recognition method based on the angular radial signature (ARS), extracted face features with Kernel principal component analysis (KPCA), and realized face recognition with support vector machine. The experiment suggested that the error rate of the method was smaller than 1%, which verified the reliability of the method. Deep learning method also has a good application in face detection and recognition [11]; convolutional neural network (CNN), especially, has excellent performance in the field of image recognition. At present, methods such as Alexnet [12], VGGnet [13], FaceNet [14] have been developed. In this study, CNN-based face detection and recognition method was studied. CNN-based face detection model, self-learning CNN face recognition model and Spatial Pyramid Pooling (SPP)-Net face recognition model were introduced respectively. The model was tested using LFW data set to understand the effectiveness of different methods.

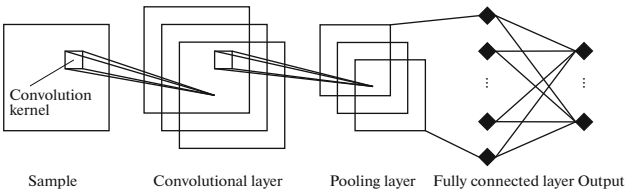

Fig. 1. The structure of CNN.

2. CONVOLUTIONAL NEURAL NETWORK

2.1. Composition of CNN

CNN [15] consists of feature extraction module and classifier. There are convolutional layer and pooling layer in each feature extraction module, and classifier includes 1–2 fully connected layer. A simple CNN structure is shown in Fig. 1.

The convolutional layer extracts image features by convolution operation, and convolutional calculation was performed on any point of an image to get image features by sliding of convolution kernel. The more convolution layers, the better image feature extraction ability. If the first layer is a convolutional layer, then the computational formula is:

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} \otimes k_{ij}^l + b_j^l\right), \tag{1}$$

where x_j^l stands for output after convolution, l stands for the number of current layer, f stands for activation function, M_j stands for input image set, x_i^{l-1} stands for output feature map matrix, $\otimes$ stands for convolution operation, k stands for matrix of convolutional kernel weight value, and b stands for biasing.

Pooling layer is generally located after convolutional layer, which is used for aggregating different characteristics of image to reduce characteristic number and data dimension. Pooling technology includes mean pooling and max pooling. The computational formula of max pooling is:

$$x_j^l = f\left(\beta^l \times D\left(x_i^{l-1}\right) + b^l\right), \tag{2}$$

where D stands for pooling technology.

After multi-layer convolution, image features extracted by convolutional layer are classified using fully-connected layer. If the l-th layer is a fully-connected layer, the computational formula is:

$$\delta_j^l = f\left(\sum_{i=1}^{n} x_i^{l-1} \otimes \varepsilon_{ij}^l + b_j^l\right), \tag{3}$$

where n stands for number of neurons and ε_{ij}^l stands for the strength of connection between neurons.

2.2. Activation Function

To enhance the expression ability of neural network, the commonly used activation functions in CNN are all non-linear activation functions, including:

(1) Sigmoid function is:

$$f(x) = \frac{1}{(1+e^{-x})}, \quad f'(x) = f(x)[1-f(x)]. \tag{4}$$

(2) Tanh function is:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\text{Sigmoid}(2x) - 1. \tag{5}$$

(3) ReLu function is:

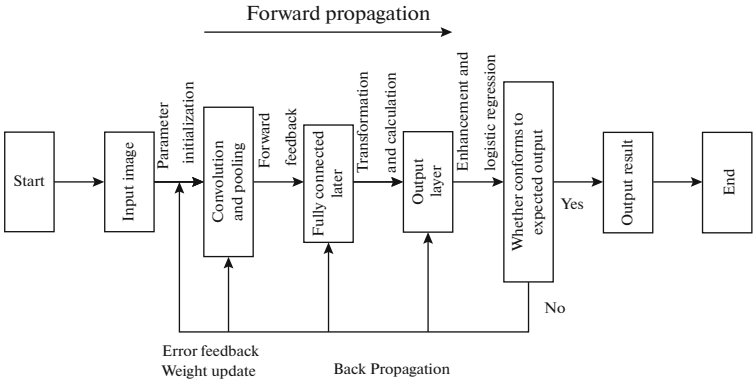

Fig. 2. The process of CNN training.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases} \tag{6}$$

(4) Leaky-ReLu function is:

$$f(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0, \end{cases} \tag{7}$$

where a is a very small real number, usually 0.01.

2.3. Softmax Classification

CNN generally classifies using Softmax classifier. For input x, the probability of $y = i$ is:

$$p(y = i|x;\theta) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_i^T x}}. \tag{8}$$

The probability of k classification of x is:

$$g_\theta\left(x^{(x)}\right) = \begin{bmatrix} p\left(y^{(i)} = 1 \middle| x^{(i)};\theta\right) \\ p\left(y^{(i)} = 2 \middle| x^{(i)};\theta\right) \\ \cdots \\ p\left(y^{(i)} = k \middle| x^{(i)};\theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \cdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}, \tag{9}$$

where $g_e(x)$ stands for hypothesis function and θ_i stands for model parameter.

2.4. CNN Training

CNN training includes forward propagation and back propagation, as shown in Fig. 2.

After the sample set is input, the parameters of the network are initialized, and then the image features are extracted and fed back to the fully connected layer. After transformation and calculation, the data are enhanced and classified, and the actual output is compared with the expected output. It is output if satisfying the expectation; otherwise the network will be back-propagated, the weight value will be updated through minimizing error, and the result is calculated again.

3. CNN BASED HUMAN FACE DETECTION MODEL

CNN designed for human face detection in this study include 3 convolutional layers, 4 pooling layers, introduction layer and 3 fully connected layers. The parameters of different layers are shown in Table 1.

Table 1. Parameters of CNN for human face detection

Layer	Input	Convolution kernel	Output
Convolutional layer 1	96 × 96 × 3	9 × 9 × 3	88 × 88 × 32
Pooling layer 1	88 × 88 × 32	2 × 2 × 1	44 × 44 × 32
Convolutional layer 2	44 × 44 × 32	9 × 9 × 32	36 × 36 × 32
Pooling layer 2	36 × 36 × 32	2 × 2 × 1	18 × 18 × 32
Convolutional layer 3	18 × 18 × 32	9 × 9 × 32	10 × 10 × 32
Pooling layer 3	9 × 9 × 32	3 × 3 × 1	3 × 3 × 32
Pooling layer 4	3 × 3 × 32	3 × 3 × 1	1 × 1 × 32
Introduction layer	1 × 1 × 3232		1 × 1 × 3232
Fully connected layer 1	1 × 1 × 3232		1 × 1 × 500
Fully connected layer 2	1 × 1 × 500		1 × 1 × 100
Fully connected layer 3	1 × 1 × 100		1 × 1 × 1

Convolution operation was performed according to the method described in section 2.1. The activation function used was ReLu function. The step length of convolution kernel was 1 × 1.

To effective extract image features, max pooling was performed on the first three pooling layers to obtain the local texture information of images:

$$x_j^l = \max\left(M_j^{l-1}\right). \tag{10}$$

Average pooling was performed on the last pooling layer to obtain the global information of images:

$$x_j^l = \text{mean}\left(M_j^{l-1}\right). \tag{11}$$

The activation function of fully connected layer 1 and 2 was ReLu function, and the activation of fully connected layer 3 was Logistic regression function:

$$f(x) = \begin{cases} 1 & \dfrac{1}{1+e^{-x}} \geq 0.5 \\ 0 & \dfrac{1}{1+e^{-x}} < 0.5. \end{cases} \tag{12}$$

The learning algorithm of CNN was stochastic gradient descent algorithm, and the objective function was:

$$J(W) = \frac{1}{2}\sum_{i=1}^{N}\left(f^i(W) - d^i\right)^2, \tag{13}$$

where N stands for number of samples, $f^i(W)$ stands for the output of CNN, and d^i stands for the classification label of sample, among which positive sample was labeled as 1 and negative sample was labeled as 0.

4. CNN BASED HUMAN FACE IDENTIFICATION MODEL

4.1. Self-learning CNN Model

A five-layer initial network structure was designed, including convolutional layer C1 and C2, pooling layer S1 and S2 and fully connected layer. The weight value was updated using back propagation during training. The network convergence speed was calculated:

$$\text{error}_{\text{hope}} - \text{error}_{\text{real}} \geq T, \tag{14}$$

where T stands for expected threshold, 0.1. The average error of system was:

$$\text{error}_{\text{real}} = \left\| O - O_{\text{lab}} \right\|^2 = \frac{\sum_{j=1}^{N}\sum_{i=1}^{m}\left(o_i^j - o_{\text{lab}}^j\right)^2}{N}, \tag{15}$$

where N stands for the total number of samples, m stands for the number of output categories, o_i^j stands for the output of neuron i corresponding to sample j, o_{lab}^j stands for the real category label, and O and O_{lab} stand for binary matrices of 0 and 1 with $m \times N$.

When training network could not reach convergence, global extended learning which meant expanding new branch on the basis of initial network structure was needed. The initial branch was set as A, and the new branch was set as B; the output result of network at this moment was:

$$o = f\left(o_A + w_B o_B\right), \tag{16}$$

where o_A and o_B stand for binary vectors of $m \times 1$ and w_B stands for column vector of m dimension. Back propagation algorithm was used for training to complete global extended learning.

After global extended learning, network might not reach the optimal state. At this moment, new local branch was added for local extended learning. Local extended learning meant integrating networks after global extended learning and then implementing convolution and fusion computation. The feature map of pooling layer S1 was input, then we have:

$$\text{Cl}_{\text{local}} = f\left(\text{Sl}_A \otimes k_A + \text{Sl}_B \otimes k_B\right), \tag{17}$$

where Cl_{local} stands for the feature map of C1 in local branch, Sl_A and Sl_B stand for the feature map of branch A and B in S1, and k stands for convolution kernel.

The local and global output were superposed, and the output result of network at this moment was:

$$o = f\left(y_{\text{global}} + w_{\text{local}} y_{\text{local}}\right), \tag{18}$$

where y_{global} stands for global output, i.e., the output of global extended learning, w_{local} stands for column vector of local branch, and y_{local} stands for local output, i.e., the output of local extended learning.

After global and local extended learning, the network structure reached a very high precision, and a self-learning CNN model was obtained.

4.2. SPP-Net Model

SPP can aggregate and transform features obtained in the process of convolution and pooling and input them into fully-connected layer. There is no requirement for the input size of image. Fully-connected layer can be transformed into a new feature vector required by the fully-connected layer after the pooling operation of SPP. SPP-Net model uses pyramid pooling in the last layer of CNN and output by using three feature matrices with different scales. If the image feature obtained after convolution was $a \times b$, sampling was performed using three windows with different sizes, w, and step length s. The computational formulas are:

$$w = a \times b, \quad \left\lceil \left(\frac{a}{n}\right) \times \left(\frac{b}{n}\right) \right\rceil, \tag{19}$$

$$s = n \times n, \quad \left\lfloor \left(\frac{a}{n}\right) \times \left(\frac{b}{n}\right) \right\rfloor, \tag{20}$$

where $\lceil\ \rceil$ stands for round up to an integer and $\lfloor\ \rfloor$ stands for round down to an integer. The output characteristic matrices after SPP were 1×1, 2×2 and 4×4. Suppose there was k feature maps, then the output matrices were $1 \times k, 4 \times k, 16 \times k$. Then they were arranged into $(21 \times k) \times 1$ feature vectors.

The structure of SPP-Net model is as follows.

(1) Input layer: The preprocessed image X was input.

(2) Convolutional layer C1: Convolution was performed on the input image through 15 convolution kernel in a size of 5×5. ReLu function was used. Then 15 feature maps were obtained:

$$X_j^{c1} = \text{Re}\,LU\left(X \otimes K_{ij}^{c1} + b_j^{c1}\right) = \max\left(0, X \otimes K_{ij}^{c1} + b_j^{c1}\right), \quad i = 1, \quad j = 1,2,\ldots,15. \tag{21}$$

Fig. 3. Pictures of human face in LFW data set.

Fig. 4. Pictures of human face after preprocessing.

(3) Pooling layer S2: Max pooling was performed. The pooling window was 2 × 2, and the step size was 2. After sampling, the size of feature map $X_i^{S2}, i = 1,2,\ldots,15$ was 28 × 28.

(4) Convolutional layer C3: Convolution was performed on X_i^{s2}. Thirty feature maps in C3 layer were connected with 15 feature maps in the upper layer. The convolution kernel was 5 × 5, and the size of feature map after convolution was 24 × 24.

(5) Pooling layer S4: Max pooling was performed. The pooling window was 2 × 2, and the step size was 2. After sampling, the size of feature map $X_i^{S2}, i = 1,2,\ldots,15$ was 12 × 12.

(6) Convolutional layer C5: Convolution was performed on X_i^{s2}. Sixty feature maps in C5 layer were completely connected with the upper layer. The convolution kernel was 5 × 5, and the size of feature map after convolution was 5 × 5:

$$X_j^{C5} = \max\left(0, \sum X_i^{s4} \otimes K_{ij}^{c5} + b_j^{c5}\right), \quad i = 1,2,\ldots,30, \quad j = 1,2,\ldots,60. \tag{22}$$

(7) SPP6: Max pooling was performed on sixty feature maps in C5 using three scales to obtain 1260 × 1 column vector, and it was input into the fully-connected layer.

(8) Input layer FC7: The output x^{FC7} of the fully-connected layer was input into Softmax classifier. The vector y_{output} was calculated, and the image was classified.

5. ALGORITHM TESTING

5.1. Experimental Data and Preprocessing

The training and testing of the model was realized using Caffe framework. The human face detection and recognition model was tested using LFW data set. LFW data set included 13233 pictures, most of which were color pictures. The faces in the images had different expressions, gestures, illumination and shields, which could effectively test the model. Part of the pictures in LFW data set is shown in Fig. 3.

To obtain better human face pictures, the pictures needed to be preprocessed. The pictures after processing are shown in Fig. 4.

Table 2. Human face detection result

	Number of actual positive samples	Number of actual negative samples	Overall accuracy rate
Number of positive samples detected	986	6	99%
Number of negative samples detected	14	994	
Accuracy rate	98.6%	99.4%	

Table 3. Human face recognition result

		Number of actually matched samples	Number of actually unmatched samples	Overall accuracy rate
Self-learning CNN model	Number of samples detected as matched	956	58	94.9%
	Number of samples detected as unmatched	44	942	
Accuracy rate		95.6%	94.2%	
SPP-Netmodel	Number of samples detected as matched	921	64	92.85%
	Number of samples detected as unmatched	79	936	
Accuracy rate		92.1%	93.6%	

5.2. Testing of the Human Face Detection Model

One thousand pictures of human face were selected from LFW data set randomly for testing of the human face detection model. Picture of human face in resolution of 96 × 96 was taken as the positive sample. Then an image block in the same size was randomly cut from the pictures as negative samples. The testing results of the model is shown in Table 2.

Table 2 shows that 986 positive samples and 994 negative samples were correctly detected by the model in the detection of 1000 positive samples and 1000 negative samples. The detection accuracy rate was 98.6 and 99.4% respectively, and the overall accuracy was 99%. It showed that the CNN face detection model designed in this study had a high accuracy rate.

5.3. Test of the Human Face Recognition Model

One thousand pairs of matched face samples and one thousand pairs of unmatched face samples were selected from LFW data set to test the face recognition model.

Table 3 shows the result of face recognition using different convolution neural network models. It was found that 956 pairs of matched samples and 942 pairs of matched samples were identified by using the self-learning CNN model, and the overall accuracy rate reached 94.9%; when the SPP-Net model was used, 921 pairs of matched samples were identified, and 936 pairs of unmatched samples were identified, with an overall accuracy rate of 92.85%. Therefore for the same sample, the accuracy rate of the self-learning CNN face recognition model was higher than that of the SPP-Net model, which showed that the self-learning CNN model was more effective in face recognition.

6. DISCUSSION AND CONCLUSIONS

With the development of society, more and more attention has been paid to face detection and recognition. Identity recognition is of great importance to individual information and security. Face recognition is the most convenient and intuitive method in identity recognition. It judges a person's identity information by comparing face with known face. It can acquire information in natural state and has great advantages compared with fingerprint recognition and iris recognition [16].

At present, face recognition has played an important role in many fields. For example, in the criminal field, criminals can be identified through face recognition technology; in the financial field, user intelligent and secure identity authentication can be realized through identity recognition; in the field of human-computer interaction, encryption and decryption of personal computers, mobile phones, etc. can be realized through face recognition technology [17], which can enhance security; in the field of security, face recognition technology can be used to monitor public places and prevent crime [18] and can also be used for strengthening the security of communities, companies and other areas.

Traditional face recognition methods include geometric feature method and local feature method. With the development of technology, people find that convolutional neural network has great potential in image recognition. It is applied to face recognition and many detection and recognition models have been

developed. The research of convolutional neural network in face recognition field will become more and more important.

Firstly, the structure of CNN was briefly introduced, and then the face detection model was designed. In face recognition, two kinds of face recognition models were introduced. One was to add global and local extended learning on the basis of CNN to improve the accuracy of the network and form a self-learning CNN model. The other was SPP-Net model which combined spatial pyramid pooling, and the last layer of the CNN model used spatial pyramid pooling. Finally, the face detection and recognition model was tested by the face pictures from LFW dataset. After pretreatment, the pictures were input into the models. The results showed that the accuracy rate of the designed face detection model was 98.6% for positive samples and 99.4% for negative samples, and the overall accuracy rate was 99%. It indicated that the face detection model had a good accuracy rate and could effectively detect face in pictures. In the face recognition model test, the accuracy rate of the self-learning CNN model was 94.9% and that of the SPP-Net model was 92.85%, demonstrating that the self-learning CNN model was better than the SPP-Net model in face recognition.

In summary, convolutional neural network has a great value in the field of face detection and recognition. Although the accuracy rates of different face detection and recognition models were different, they all have high reliability and can detect and recognize faces accurately, which is worth further research and promotion in practice.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

1. Ghiass, R.S., Arandjelovic, O., Bendada, H., et al., Infrared face recognition: A comprehensive review of methodologies and databases, *Pattern Recognit.,* 2014, vol. 47, no. 9, pp. 2807–2824.
2. He, R., Wu, X., Sun, Z., et al., Wasserstein CNN: Learning invariant features for NIR-VIS face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, no. 99, p. 1.
3. Hu, S., Choi, J., Chan, A.L., et al., Thermal-to-visible face recognition using partial least squares, *J. Optic. Soc. Am. A Opt. Image Sci. Vision*, 2015, vol. 32, no. 3, pp. 431–442.
4. Peng, Y., Wang, S., Long, X., et al., Discriminative graph regularized extreme learning machine and its application to face recognition, *Neurocomputing*, 2015, vol. 149, pp. 340–353.
5. Shi, X., Yang, Y., Guo, Z., et al., Face recognition by sparse discriminant analysis via joint L2,1-norm minimization, *Pattern Recognit.*, 2014, vol. 47, no. 7, pp. 2447–2453.
6. Prakash, N., and Singh, Y., Fuzzy support vector machines for face recognition: A review, *Int. J. Comp. Appl.*, 2015, vol. 131, no. 3, pp. 24–26.
7. Ding, C., Choi, J., Tao, D., et al., Multi-directional multi-level dual-cross patterns for robust face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, vol. 38, no. 3, pp. 518–531.
8. Lai, J., Wang, Y., Zhou, G., et al., A fast (l)1-solver and its applications to robust face recognition, *J. Ind. Manage. Optim.,* 2017, vol. 8, no. 1, pp. 163–178.
9. Zhang, L., Zhou, W.D., and Li, F.Z., Kernel sparse representation-based classifier ensemble for face recognition, *Multimedia Tools Appl.*, 2015, vol. 74, no. 1, pp. 123–137.
10. Lei, Y., Bennamoun, M., Hayat, M., et al., An efficient 3D face recognition approach using local geometrical signatures, *Pattern Recognit.*, 2014, vol. 47, no. 2, pp. 509–524.
11. Zhang, K., Zhang, Z., Li, Z., et al., Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Sign. Proc. Lett.*, 2016, vol. 23, no. 10, pp. 1499–1503.
12. Bagherinezhad, H., Rastegari, M., and Farhadi, A., LCNN: Lookup-based convolutional neural network, *IEEE Conf. Computer Vision and Pattern Recognition. IEEE Computer Society*, 2017, pp. 860–869.
13. Lavinia, Y., Vo, H.H., and Verma, A., Fusion based deep CNN for improved large-scale image action recognition, *IEEE Int. Symp. Multimedia*, San Jose, CA, 2017, pp. 609–614.
14. Schroff, F., Kalenichenko, D., and Philbin, J., FaceNet: A unified embedding for face recognition and clustering, *IEEE Conf. Computer Vision and Pattern Recognition. IEEE Computer Society*, 2015, pp. 815–823.
15. Rawat, W., and Wang, Z., Deep convolutional neural networks for image classification: A comprehensive review, *Neural Comput.*, 2017, vol. 29, no. 9, pp. 2352–2449.
16. Galbally, J., Marcel, S., and Fierrez, J., Biometric antispoofing methods: A survey in face recognition, *IEEE Access*, 2014, vol. 2, pp. 1530–1552.
17. Smith, D.F., Wiliem, A., and Lovell, B.C., Face recognition on consumer devices: Reflections on replay attacks, *IEEE Trans. Inf. Forensics Secur.*, 2015, vol. 10, no. 4, pp. 736–745.
18. Kang, D., Han, H., Jain, A.K., et al., Nighttime face recognition at large standoff: Cross-distance and cross-spectral matching, *Pattern Recognit.*, 2014, vol. 47, no. 1, pp. 3750–3766.