

## Recognition of Motion-blurred CCTs based on Deep and Transfer Learning

Yun Shi<sup>[1,2]</sup>, Yanyan Zhu<sup>[2,\*]</sup>

[1] College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016 China.

[2] School of Electronics and Information Engineering, West Anhui University, Lu'an, 237012, China.

[\*] Corresponding Author: Yanyan Zhu: 710667612@qq.com

**Abstract** This paper uses deep and transfer learning in identifying motion-blurred Chinese character coded targets (CCTs) to reduce the need for a large number of samples and long training times of conventional methods. Firstly, a set of CCTs are designed, and a motion blur image generation system is used to provide samples for the recognition network. Then, the OTSU algorithm, the expansion, and the Canny operator are performed on the real shot blurred images, where the target area is segmented by the minimum bounding box. Next, a sample is selected from the sample set according to the 4:1 ratio, i.e., training set: test set. Furthermore, under the Tensor Flow framework, the convolutional layer in the AlexNet is fixed, and the fully-connected layer is trained for transfer learning. Finally, numerous experiments on the simulated and real-time motion-blurred images are carried out. The results showed that network training and testing take 30 minutes and two seconds on average, and the recognition accuracy reaches 98.6% and 93.58%, respectively. As a result, our method achieves higher recognition accuracy, does not require a large number of samples for training, requires less time, and can provide a certain reference for the recognition of motion-blurred CCTs.

**Keywords:** Chinese character coded targets (CCTs), Deep learning, Image recognition, Motion blur, Transfer learning.

## 1 Introduction

The three-dimensional (3D) morphology of high-speed blades needs accurate measurement. Although there are edge detection and shape recognition algorithms for regular size static visual features [1, 2], these algorithms are not suitable for visual feature recognition with motion blur effects. Although increasing the camera frame rate and shortening the exposure time can obtain images without obvious motion blur effects, high-speed cameras are expensive, and short exposure times may reduce the image signal-to-noise ratio (SNR) [3], which affects the accuracy of the 3D image measurement. In [4], it is shown that the algorithms using multiple blurry high SNR images perform significantly better than the ones using high-noise images with high speed.

The movement of a target during the exposure causes a certain degree of blur, which is an important factor affecting the accuracy of the 3D vision measurement. The motion blur effect should not be considered as the degradation of the image but should be treated as the result of the superimposed imaging of the 3D target on the motion path during the exposure. The authors of [5-8] revealed the 3D geometric reconstruction of moving targets and the quantitative relationships between the motion paths and the image blur effects from different sides.

Machine vision-based measurement has the advantages of non-contact, good real-time performance, and high flexibility. It can detect, recognize, and quickly measure objects. The research on the 3D vision measurement of the motion blur effect images needs to solve the problem of accurate recognition of active visual features of motion blur effect first. Currently, there are a few related studies at home and abroad. The authors of [9] achieved high recognition accuracy for image recognition based on convolutional neural networks. However, their model requires a large number of training samples and a long training time.

In recent years, deep learning has received significant attention in the field of artificial intelligence. Deep learning forms a denser high-level semantic abstraction by combining low-level features. It can automatically discover the distributed feature representation of data and solve the problem of artificially designing features in the traditional machine learning [10]. Deep learning has made significant progress in several areas [11, 12]. For example, AlexNet, the conventional deep learning model for image classification, won the ILSVRC2012 competition. In [13], it is shown that the convolutional layer of the AlexNet around the data layer took more than 70% of the time during training. If the parameter calculation of the convolutional layer is reduced, the training time will be greatly shortened, and the model size will be compressed.

Transfer learning is the knowledge learned from a sample to help to solve the tasks of the new related samples. The AlexNet model, trained based on large-scale samples, achieves a stronger generalization performance when it gets better classification results. At the same time, the transfer learning ability gets stronger when the image features of the new and old samples are relevant to each other.

This paper uses a pre-trained AlexNet model establishing a motion-blurred CCTs AlexNet\_Target (AT) recognition model to reduce the required number of samples and the training time. The common knowledge of image classification obtained by the AlexNet model on the image dataset is transferred to the AT recognition model using the transfer learning method.

## 2 The Proposed Methods

### 2.1 Motion-blurred CCTs

#### 2.1.1 Design of the CCTs

Fig. 1.a shows the structure of the CCT. The CCT mainly includes three concentric circles in black and white and different Chinese characters situated at the outside of the circles. The center of the concentric circles is the position of the CCT. Fig. 1.b shows a CCT "非" generated by computer simulations based on C++. The circles are of 4mm, 8mm, and 12mm of diameters, and the size of the CCTs is 22×22mm.

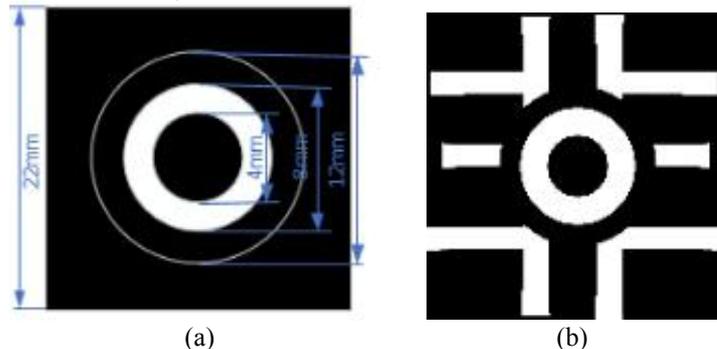


Figure 1. Structure and dimensions of the CCT.

One hundred Chinese characters are used as coding targets. The characters in each code target are different; thus, they can be used as unique target characterization points. They are numbered as 0, 1, ..., 99, which are used as corresponding code values. For the clear coding targets shown in Fig. 1.b, coding values can be obtained through the Canny operators, so that each coding point can be accurately identified. Then, the target points with the same name on different images are matched, and the 3D reconstruction and measurement are performed. Fig. 2 shows the actual imaging of CCT with motion blur effect. As seen, the shape of the edges and the clear image have changed significantly, which vary with the shooting angle and the degree of blur. The changes can be very different, and it is impossible to use the methods in [14] to identify the CCTs with the motion blur effect based on the clear edge images.



Figure 2. The real shot of CCT.

### 2.1.2 Generating Blur CCTs by Simulations

The widespread utilization of deep learning is limited by its need for tens of thousands of datasets. It is also time-consuming, labor-intensive, and even difficult to take a large number of motion-blurred CCTs and manually label them for network training. In this paper, the model in [9] is used to simulate the generation of motion-blurred images of CCTs in different poses and motion paths. The specific steps are as follows:

**Step 1:** Calibrate the camera, determine the internal parameter matrix of it, and record the image plane as  $\pi$ .

**Step 2:** In the camera coordinate system, the space movement area of the measured object is recorded as  $\Omega$ . Determine the side length of the square CCTs are 22mm. Set the code target serial numbers  $M = \{0, 1, \dots, 99\}$  and prepare the corresponding CCTs  $I_m, m \in M$ .

**Step 3:** The method of the virtual camera coordinate system is a virtual scene. Determine the internal parameter matrix  $K$  of the virtual camera.

**Step 4:** The virtual code target is a square with a side length of 22mm. Chinese characters  $I_m$  are coded on the front of the square, and  $I_m$  just fills the square. Set the plane of the square to  $\pi_0$ .

**Step 5:** In  $\Omega$ , randomly select the starting point  $P_0$  and the ending point  $P_1$  of the coded point movement. Two vectors are randomly selected as the starting direction vector  $\varpi_0$  and the ending direction vector  $\varpi_1$ .

**Step 6:** Using  $P_0, P_1, \varpi_0, \varpi_1$ , determine a cubic spline curve  $P = P(t), t \in [0, 1]$ . The sampling points  $t_n = n/N, n = 0, 1, \dots, N$  are divided equally on the interval (0,1), where  $N$  represents the number of static instants.

**Step 7:** Calculate the spatial homogeneous coordinates  $S_1, S_2, S_3, S_4$  of the four vertices of the square at time  $t_n$ , and the column vector  $L_i = P(t_n) + R_n S_i, i = 1, 2, 3, 4$ . Calculate the projection  $\mu_i = K L_i, i = 1, 2, 3, 4$  of  $L_i$  on the image plane.

**Step 8:** Calculate the homography matrix  $H_n$  from  $\pi_0$  to  $\pi$  according to the relationship between  $S_i$  and  $\mu_i$ . Use  $H_n$  and  $I_m$  to calculate the imaging result  $J_{mn}$  of the virtual camera at time  $t_n$ .

**Step 9:** Calculate the blurred image  $B_{mk} = (\sum_{i=k}^{k+b-1} J_{mi}) / b$ , where  $k=1, 2, \dots, n-b+1$  and  $B_{mk}$  represents the motion-blurred CCT image of the  $m$ -th code target.

Repeat from Step 5 to Step 9 to obtain a motion-blurred CCT image and a code point identity sample. This sample can be used for training and testing of the AT recognition model.

## 2.2 Recognition of Motion-blurred CCTs

### 2.2.1 Segmentation and Extraction of Motion-blurred CCTs

To realize network training and testing, the region where the CCTs are located needs to be segmented from the image. For a single motion-blurred CCT image: (i) the OTSU algorithm post-expansion processing is performed; (ii) the Canny edge detection operator is used to extract the edge contours; and (iii) the minimum bounding box is used to segment the target area of the blurred CCT. Since the actual captured image contains multiple CCTs at the same time, the difficulty of segmentation will increase. Hence, contour screening is performed on the gathered minimum bounding box area, perimeter, minimum circumscribed rectangle aspect ratio, and the average gray value. Fig. 3 shows the segmentation algorithm and the results obtained.

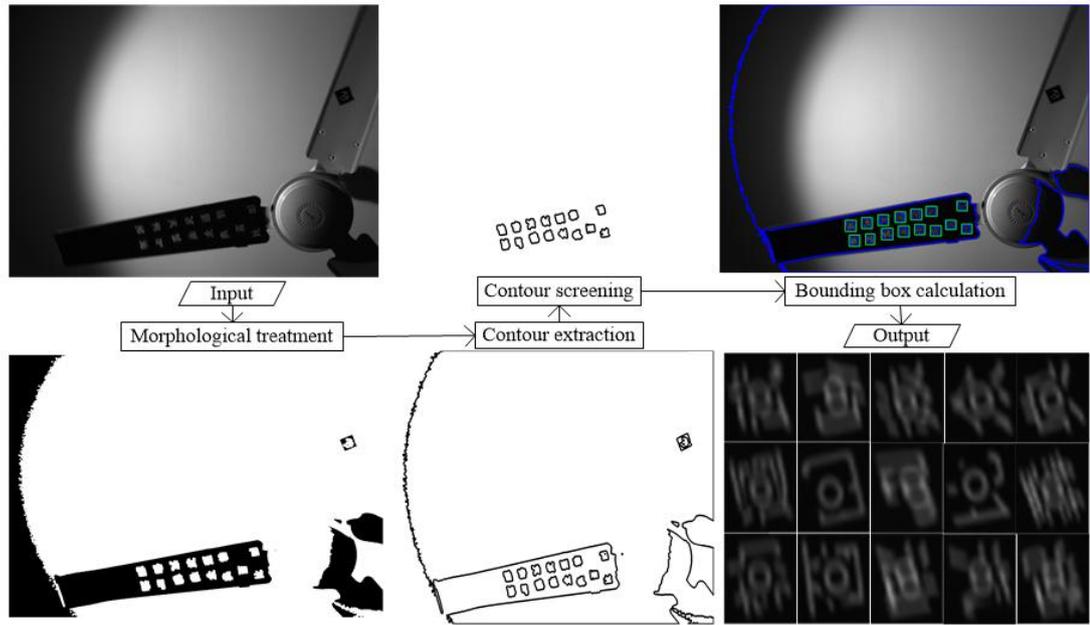


Figure 3. Segmentation algorithm and the results obtained.

### 2.2.2 Network Model

- Transfer learning

The AlexNet model is trained on large image datasets to learn several common features required for image recognition and classification. Utilizing the transfer learning idea in [15], a lot of common knowledge learned on the dataset by pre-trained models. For example, the AlexNet model is fully used for the recognition and classification of the motion-blurred CCTs. There are two methods of transfer learning. The first one is parameter transfer learning, which only initializes a few layers of the pre-trained network. The remaining layer weight parameters are directly used, and then new parameters are utilized to fine-tune the parameters. The second is feature transfer learning. After removing the pre-trained network classifier, the activation value of the previous convolution layer is put into the classifier for classification training. In this paper, the feature transfer learning method is used to convert the AlexNet model into an AT recognition model for the motion-blurred CCTs.

- Model Structure

The AlexNet model is one of the representative network models for deep learning. It consists of an input layer, a convolutional layer (C), a local response normalization layer (LRN), a pooling layer (P), a fully connected layer (F), and an output layer. The mathematical expressions of the AT recognition model designed in this paper are as follows:

$$C : h_{i,j}^k = \text{ReLU}[(\omega^k \otimes X_c)_{i,j} + b_k], \quad (1)$$

where  $h_{i,j}^k$  is the neuron value of the  $k$ -th feature map at  $(i, j)$ ,  $\omega^k$  is the weighting factor of the  $k$ -th feature map,  $X_c$  is the input of the convolutional layer,  $\otimes$  is the convolution operation, and  $b_k$  is the offset of the  $k$ -th feature map.

$$P : h_{i,j}^k = \text{Max}(x_{2i+m, 2j+n}^k \mid m = \{0, 1\}, \{0, 1\}), \quad (2)$$

where  $\text{Max}(\cdot)$  is the maximum pooling function, and  $x_{2i+m, 2j+n}^k$  is the neuron value of the  $k$ -th feature map at  $(2i+m, 2j+n)$ .

$$F : f_i = \text{PR eLU}(\omega_i^T x_F + b_i), \quad (3)$$

where  $f_i$  is the neuron value of the  $i$ -th output,  $\omega_i^T$  is the  $i$ -th neuron weight,  $x_F$  is a fully connected layer input, and  $b_i$  is the bias of the  $i$ -th neuron. The mathematical expression of PReLU [16] can be given as  $y = \max(0, \omega x + b) + a_i \min(0, \omega x + b)$ , in which  $a_i$  indicates the learning rate that prevents the complete loss

of information on the negative axis and has a correction effect. If  $a_i = 0$ , PReLU degrades to ReLU. If  $a_i$  is a small fixed value, then PReLU degrades to Leaky ReLU.

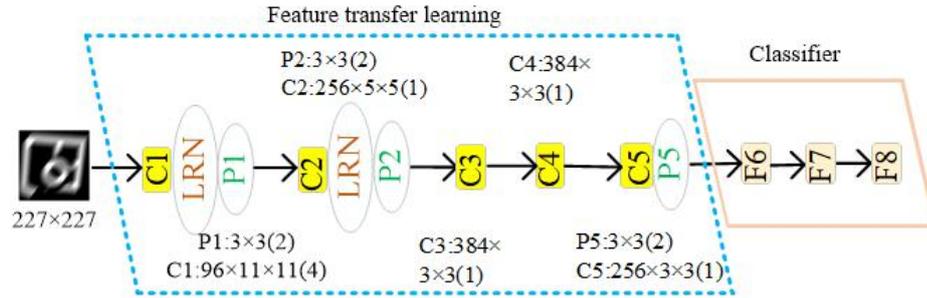


Figure 4. AT recognition model of the Chinese character "巴".

Fig. 4 shows the AT recognition model of the Chinese character "巴". The dotted frame is the feature transfer learning stage, and the solid line is the classifier. The AT recognition model has five convolutional layers (C1, C2, C3, C4, C5), three pooling layers (P1, P2, P5), and three fully connected layers (F6, F7, F8). The number of outputs of the fully connected layer F8 is the number of corresponding target categories. The input picture size is 227×227 pixels. C1: 96×11×11 (4) indicates that the C1 layer has 96 convolution kernels, and the convolution kernel window size is 11×11. The convolution kernel has a sliding step size of four on its input feature surface. P1: 3×3 (2) indicates that the kernel window size is 3×3, and the sliding step size is two. To prevent the overfitting of pooling layers to convolutional layers and fully-connected layers to output layers, the dropout technology is used. The output layer adopts Softmax [17] as the regression model, and the category with the highest probability is considered as the output category. The **cross-entropy** loss function is used as the objective function. The AT recognition model fixes the parameters of the feature transfer learning phase and trains only the classifier phase. The feature map results of the five running Chinese convolutional layers of the CCT "巴" are shown in Fig. 5. As seen, the "巴" font sign on the C1 layer is still obvious; however, it is difficult to recognize its features on the C5 layer by the naked eye.

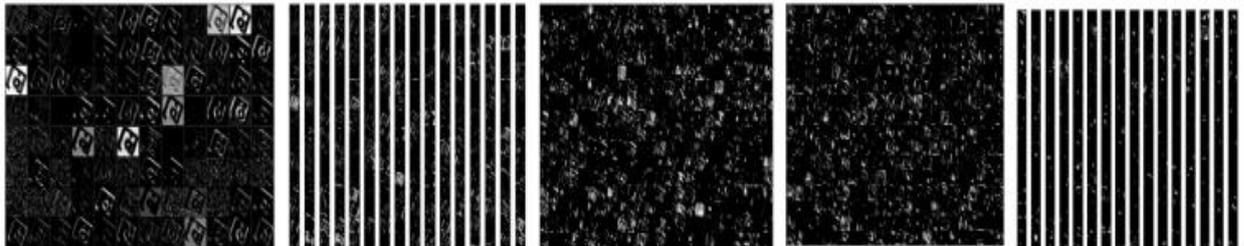


Figure 5. Feature map of the CCT "巴".

### 3 Experimental Results

#### 3.1 Simulated and Real Shot Blurred Images

We followed Section 2.1.2 to simulate the generation of 5000 motion-blurred CCTs on a DELL Precision T5610 workstation with 2GB of video memory, 24GB of RAM, and the Quadro K2000 GPU. Since it is a random simulation of various CCTs, the number of images in the dataset is approximately equal. Fig. 6 shows a partially-simulated motion-blurred CCTs image.

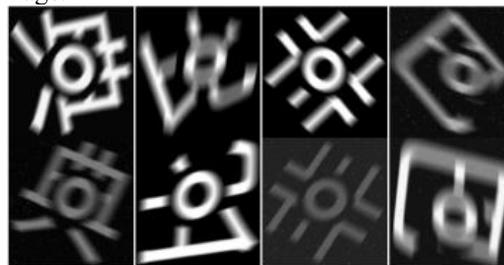


Figure 6. A partially-simulated motion-blurred CCTs image.

To capture the actual motion-blurred CCTs image, a Basler A102f camera with a resolution of  $1392 \times 1040$  pixels was used to shoot 15 types of CCTs arranged on ceiling fan blades. The ceiling fan was always in motion during the shooting process, so that each CCT could be photographed 300 times from different angles and at different speeds. Fig.7 shows some of the actual motion-blurred CCTs segmented following Section 2.2.1.

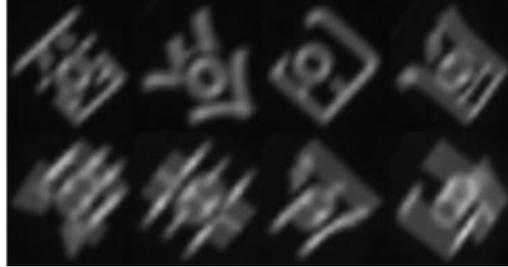


Figure 7. The real shot of the CCT image.

### 3.2 Training and Test of the Blurred Images

We used the ratio of the number of correctly identified data to the total number of data as the recognition accuracy. The AT recognition model was used to train and test 5000 blurred images generated by the simulation at a 4:1 ratio. During training, the gradient was back-propagated [18], and the optimization algorithm used a stochastic gradient descent algorithm [19]. We set the initial learning rate to 0.001 and the batch size to 50. After each update of weights and biases, the learning rate was dropped by  $10^{-6}$ . The dataset was tested after each iteration to obtain the current recognition accuracy. The average loss of the training iteration test dataset was reduced to 0.02 as the termination condition. Fig. 8 shows the relationship between the number of iterations and accuracy, where the number of iterations versus the **cross-entropy** is illustrated in Fig. 9.

The trained AT recognition model achieved 93.58% accuracy, which refers to a good generalization performance.

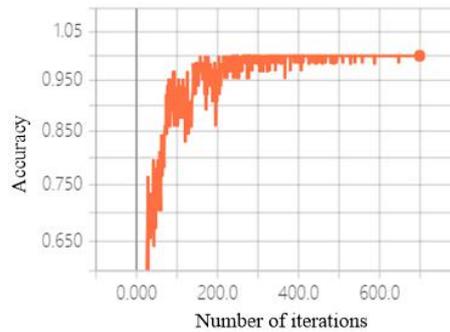


Figure 8. The relation between the number of iterations and accuracy.

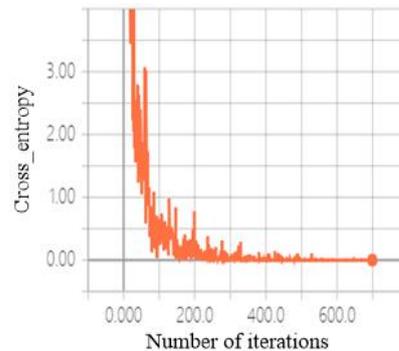


Figure 9. The relation between the number of iterations and cross-entropy.

### 3.3 Experimental Analysis

The real shot samples were compared for Multilayer Perceptron (MLP), Histogram of Oriented Gradient (HOG) + MLP, Scale-Invariant Feature Transform (SIFT) + MLP, the method in [9], and the AT recognition model in terms of recognition accuracy, training time, test time, and autonomy. The results are shown in Table 1.

Table 1. Comparison of the AT recognition model with other methods.

| Recognition methods | Accuracy (%) | Training time (h) | Test time (s) | Automatic ? |
|---------------------|--------------|-------------------|---------------|-------------|
| MLP                 | 80.5         | 1.24              | ≤1            | Yes         |
| HOG+MLP             | 80.14        | 2.15              | 12            | No          |
| HOG+MLP             | 88.64        | 18.32             | 34            | No          |
| [9]                 | 93.25        | 6.28              | 7             | Yes         |
| AT                  | 93.28        | 0.5               | 2             | Yes         |

## 4 Conclusions

This paper proposes a method for identifying the motion-blurred CCTs based on deep and transfer learning. To solve the problems of a large number of samples and the time consumed in network training, the proposed AT recognition model automatically extracts the features of motion-blurred CCTs. The recognition accuracies of the code targets for simulated and motion-blurred Chinese characters are 98.2% and 93.58%, respectively. Since transfer learning only requires 5000 new samples, the network convergence acceleration is significantly reduced from six hours to a few minutes, i.e., a great save of training time. Experimental results validated the feasibility of our method and provided a basis for 3D reconstruction and measurement under the effect of motion blur. We believe that our method is quite promising for accurate recognition of active visual features.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (No. 51575276) and the General Provincial Teaching and Research Project in Anhui Province (No. 2019jyxm0362).

## References

- [1] T.R. Fujimoto, T. Kawasaki, K. Kitamura. Canny-Edge-Detection Rankine Hugoniot conditions unified shock sensor for inviscid and viscous flows, *Journal of Computational Physics*, 2019, 396, 264-279.
- [2] M.P. Liu, W.B. Zhu, S.L. Ye. Sub-pixel edge detection based on improved Zernike moment in the small modulus gear image, *Chinese Journal of Scientific Instrument*, 2018, 8, 259-267. (in Chinese)
- [3] M. Ben-Ezra, S. Nayar. Motion deblurring using hybrid imaging. *CVPR'2003*, 657-664.
- [4] S.H. Park, M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur, *CVPR'2014*, 3366-3373.
- [5] R.W. Zhen, R.L. Stevenson. Motion deblurring and depth estimation from multiple images, *IEEE International Conference on Image Processing 2016*, 2688-2692.
- [6] L. He, G. Li, J. Liu. Joint Motion Deblurring and Super resolution from Single Blurry Image, *Mathematical Problems in Engineering 2015*, 1-10. <http://dx.doi.org/10.1155/2015/965690>
- [7] Y. Zhou, N. Komodakis. A MAP-Estimation Framework for Blind Deblurring Using High-Level Edge Priors, *European Conference on Computer Vision 2014*, 8690, 142-157.
- [8] C. Paramanand. An Rajagopalan. Shape from Sharp and Motion-Blurred Image Pair, *International Journal of Computer Vision 2014*, 107, 272-292.
- [9] M.J. Chen, H.C. Zhou, L.Y. Zhang. Recognition of Motion Blurred Coded Targets Based on Convolutional Neural Network, *Journal of Computer-Aided Design & Computer Graphics 2017*, 29, 1844-1852. (in Chinese)
- [10] L.W. Huang, B.T. Jiang, S.Y. Lv. Survey on Deep Learning Based Recommender Systems, *Chinese Journal of Computers 2018*, 41, 1619-1647. (in Chinese)
- [11] Z. Swiderska-Chadaj, H. Pinckaers, M. van Rijthoven, *et al.* Learning to detect lymphocytes in immunohistochemistry with deep learning, *Medical Image Analysis 2019*, 58, 1-12.

- 
- [12] O. Abdel-Hamid, A.R. Mohamed, H. Jiang, *et al.* Convolutional Neural Networks for Speech Recognition, *IEEE/ACM Transactions on Audio Speech & Language Processing* 2014, 22, 1533-1545.
- [13] S. Pan, Q. Yang. A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering* 2010, 22, 1345–1359.
- [14] R.S.H. Mark, W.S.E. James. A Practical Target Recognition System for Close Range Photogrammetry, *The Photogrammetric Record* 2014, 147, 337-355.
- [15] L.H. Zhong, P.X. Zong, L. Bin. Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data, *Remote Sensing* 2017, 907, 1-21.
- [16] A. Beke, T.F. Kumbasar. Learning with Type-2 Fuzzy activation functions to improve the performance of Deep Neural Networks, *Engineering Applications of Artificial Intelligence* 2019, 85, 372-384.
- [17] B.T. Li, D.C. Pi. Learning deep neural networks for node classification, *Expert Systems with Applications* 2019, 137, 324-334.
- [18] M.J. Embrechts, B.J. Hargis, J.D. Linton. Augmented Efficient Backprop for backpropagation learning in deep autoassociative neural networks, *International Joint Conference on Neural Networks* 2010, 1-6.
- [19] I. Goodfellow, Y. Bengio, A. Courville. Deep learning, *Cam-bridge: MIT Press* 2016.
-