Hand-drawn electronic component recognition using deep learning algorithm

Haiyan Wang and Tianhong Pan*

School of Electrical Engineering and Automation,
Anhui University,
Hefei, Anhui 230601, China
and
School of Electrical and Information Engineering,
Jiangsu University,
Zhenjiang, Jiangsu 212013, China
Email: hyhywang_x@163.com
Email: thpan@ahu.edu.cn
*Corresponding author

Mian Khuram Ahsan

School of Electrical and Information Engineering,
Jiangsu University,
Zhenjiang, Jiangsu 212013, China
Email: myblackeyes4u@qq.com

Abstract: Hand-drawn circuit recognition plays an increasingly important role in circuit design work and electrical knowledge teaching. Hand-drawn electronic component recognition is an indispensable part of hand-drawn circuit recognition. Accurate electronic component recognition ensures accurate circuit recognition. In this paper, a hand-drawn electronic component recognition method using a Convolutional Neural Network (CNN) and a softmax classifier is proposed. The CNN composed of a convolutional layer, an activation layer and an average-pooling layer is designed to extract features of a hand-drawn electronic component image. The kernel function for the CNN is obtained by a sparse auto-encoder method. A softmax classifier is trained for classification based on the features extracted by the CNN. The recognition method can identify rotating electronic components because of the added rotated image and achieve 95% recognition accuracy.

Keywords: electronic component recognition; CNN; convolutional neural network; aparse auto-encoder.

Reference to this paper should be made as follows: Wang, H., Pan, T. and Ahsan, M.K. (2020) 'Hand-drawn electronic component recognition using deep learning algorithm', *Int. J. Computer Applications in Technology*, Vol. 62, No. 1, pp.13–19.

Biographical notes: Haiyan Wang is a PhD student in Control Science and Engineering at Jiangsu University. Her research interests include machine learning, advance process control and controller performance monitoring.

Tianhong Pan is a Professor with the School of Electrical Engineering and Automation, Anhui University. He received his PhD degree in Control Theory and Control Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007. His research interests include multiple model approach and its application, machine learning, virtual metrology, predictive control and run-to-run control theory and practice, etc.

Mian Khuram Ahsan is a PhD student in Electrical and Electronics at Jiangsu University. He received his Master's degree from ABASYN University Peshawar in 2014. His research interests include machine learning, micro-grid system, multi-agent algorithm, and wireless communication and sensor networks.

Copyright © 2020 Inderscience Enterprises Ltd.

1 Introduction

Drawing a circuit by hands allows circuit designers to focus entirely on circuit design without being plagued by the problem of retrieving electronic components, thereby focusing on optimising circuit structures. However, rewriting a hand-drawn circuit to a computer is a tedious process. Nowadays, the proliferation of many pattern recognition algorithms (Deng et al., 2018; Hamdad and Hammouche, 2018) has brought great convenience to production and life. In addition, the success of hand-drawn circuit schematic system simulation shown in Angadi and Naika (2014) brings great convenience and infinite possibilities to circuit design. For this reason, the exploration of the automatic recognition method of hand-drawn circuits has never stopped. Edwards and Chandran (2000) developed a machine identification method for hand-drawn circuit diagrams and divided the circuit diagram recognition into three steps: segmentation, component identification, and node identification. Many scholars have proposed various circuit recognition methods based on the three-step method. Gennari et al. (2005) proposed a component extraction method based on ink density and stroke variation. Liu and Xiao (2013) proposed an image topology-based approach to replace the sliding window method in the segmentation step, significantly improving the efficiency of decomposing circuit segments into individual electronic components. In addition, Patare and Joshi (2016) also used the Fourier descriptors method to extract the characteristics of electronic components and used the Support Vector Machine (SVM) to classify for digital circuit recognition. The Fourier descriptor analysed image points in the frequency domain, the low frequency components contained general feature information, and the high-frequency components included finer details. The SVM (Chen et al., 2017) method was used as a supervised learning model for classification by finding the best hyperplane between the closest samples of two different categories. With the development of deep learning technology (Yao and Wang, 2018) and artificial neural network (ANN) technology (Sharafian and Ghasemi, 2017; Di et al., 2018), ANNs are also used for circuit recognition by Wang et al. (2001) and Rabbani et al. (2016).

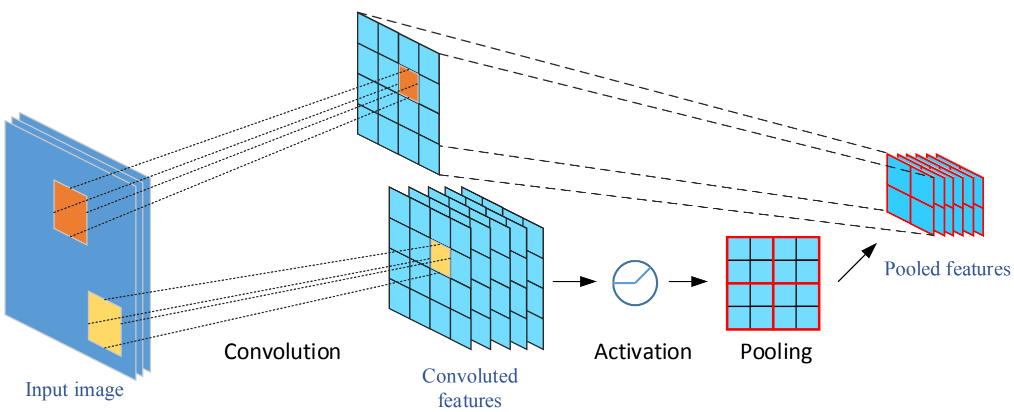
Convolutional Neural Network (CNN) is another popular deep learning method. It is an effective method for extracting multidimensional array features, such as a colour image composed of three layers of two-dimensional arrays (LeCun et al., 2015). The basic design principles of CNN come from neuroscience, which may be the most successful case of biologically inspired artificial intelligence as shown in Goodfellow et al. (2016). As early as the 1990s, CNN was first used for speech recognition and document reading by LeCun et al. (1998) and Waibel et al. (1990). In recent years, Khuwaja (2003) and He et al. (2018) used CNN for face recognition, while local and SVM-based face recognition method (Tao et al., 2016) and DRNLGBP algorithm (Wang et al., 2018) are provided to be robust to perspectives and other real-world factors. And many different kinds of image identification method are developed. Guo et al. (2016) used CNN to extract features from street view and identify the house number. The deep CNN was developed by Shin et al. (2016) to distinguish diseases based on medical images and the CNN structure was also used by Su et al. (2015) to identify 3D images from a set of rendered views on a 2D image.

The main contribution of this work is to propose an electronic component recognition method combined CNN with softmax classifier. Compared with the SVM classification, the proposed method can classify data into multiple independent categories directly and identify rotated components. A CNN presented in Ng et al. (2010) for feature extraction of hand-drawn electronic components is described in Section 2. The classification method is mentioned in Section 3. Data set generation is given in Section 4. Experimental results and analyses are presented in Section 5.

2 Feature extraction

A CNN is a feedforward structure which can extract features from the original image. Vedaldi and Lenc (2015) divided it into four steps: convolution, activation, subsampling, and full connectivity. One layer CNN is used in this paper which contains convolution, activation and subsampling (shown in Figure 1).

Figure 1 The CNN structure used in this paper



The input image is convolved with a kernel function in the convolution layer to extract feature maps from the input image. The image contains three channels, and define the input data as x in one channel, the kernel function as *K*, and the integration variables as p, q, and then the convolutional result is obtained by the convolution operation:

$$\begin{aligned} S(i,j) &= (x * K)(i,j) \\ &= \sum_p \sum_q x(p,q) K(i-p, j-q) \end{aligned} \tag{1}$$

The convolutional features are obtained by adding up the convolution results of the three channels.

The convolution kernel function is the weight of the features extracted from randomly selected small patches in the training data set. To obtain representative local features, the feedback fully connected sparse autoencoder (Ng et al., 2010) is used here.

A sparse autoencoder consists of an encoder and a decoder. An encoder obtains the hidden layer of the input data with weights $W^{(2)}$:

$$\begin{aligned} z^{(2)} &= W^{(2)} x + b^{(2)}, \\ h^{(2)} &= f\left(z^{(2)}\right), \end{aligned} \tag{2}$$

where $f(\cdot)$ is usually defined as a sigmoid function. The decoder decodes the features by $W^{(3)}$ to receive the input estimate $\hat{x}$:

$$\begin{aligned} z^{(3)} &= W^{(3)} h^{(2)} + b^{(3)}, \\ \hat{x} &= f\left(z^{(3)}\right) \end{aligned} \tag{3}$$

Define the cost function with a sparsity constraint as:

$$\begin{aligned} J_{sparse}(W,b) &= \left[\frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \|x - \hat{x}\|^2 \right) \right] \\ &+ \frac{\lambda}{2} \left(\sum_{i=1}^{s_1} \sum_{j=1}^{n} \left(W_{ij}^{(1)} \right)^2 + \sum_{i=1}^{n} \sum_{j=1}^{s_1} \left(W_{ij}^{(2)} \right)^2 \right) \\ &+ \beta \sum_{j=1}^{s_1} \mathrm{KL}\left(\rho \,\|\, \hat{\rho}_j \right) \end{aligned} \tag{4}$$

where m is the number of train images, s_1 is the number of features, λ is weight decay parameter. $\beta \sum_{j=1}^{s_1} \mathrm{KL}\left(\rho \,\|\, \hat{\rho}_j \right)$ is used to apply a sparsity constraint on the hidden units, constraining the hidden units to be inactive most of the time. β is the weight of sparsity penalty term. The average activation of hidden unit j is $\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} \left[h_j^{(2)} \right]$. ρ is a sparsity parameter and is usually a small value close to zero. $\mathrm{KL}(\cdot)$ is the Kullback-Leibler (KL) divergence, which can ensure $\hat{\rho}_j = \rho$: $\mathrm{KL}\left(\rho \,\|\, \hat{\rho}_j \right) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$.

The features and weights are determined by optimising the parameters $W^{(2)}, b^{(2)}, W^{(3)}, b^{(3)}$ based on the cost function.

The gradient descent updates method and the backpropagation algorithm are used to optimise parameters. Define the input estimate as the 3rd layer, and for the *i*-th input set:

$$\delta^{(3)} = \frac{\partial}{\partial z^{(3)}} \frac{1}{2} \|x - \hat{x}\|^2 = -(x - \hat{x}) \cdot f'\left(z^{(3)}\right) \tag{5}$$

Define the hidden layer as the 2nd layer, and for the *i*-th input set:

$$\begin{aligned} \delta^{(2)} &= \frac{\partial}{\partial z^{(2)}} J_{sparse}(W,b) \\ &= \left(\left(W_{ji}^{(3)} \right)^T \delta_j^{(3)} + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) \right) f'(z^{(2)}) \end{aligned} \tag{6}$$

Then the derivative of the cost function with respect to $W^{(2)}, b^{(2)}, W^{(3)}, b^{(3)}$ are obtained as:

$$\begin{aligned} &\nabla_{W^{(3)}} J_{sparse}(W,b) = \delta^{(3)} \left(h^{(2)} \right)^T, \\ &\nabla_{W^{(2)}} J_{sparse}(W,b) = \delta^{(2)} x^T, \\ &\nabla_{b^{(3)}} J_{sparse}(W,b) = \delta^{(3)}, \\ &\nabla_{b^{(2)}} J_{sparse}(W,b) = \delta^{(2)}. \end{aligned} \tag{7}$$

The iteration of gradient descent updates following the direction to minimize the cost function $J_{sparse}(W,b)$:

$$\begin{aligned} W^{(l)} &= W^{(l)} - \alpha \nabla_{W^{(l)}} J_{sparse}(W,b), \\ b^{(l)} &= b^{(l)} - \alpha \nabla_{b^{(l)}} J_{sparse}(W,b), \end{aligned} \tag{8}$$

where l is 2 or 3, and α is the learning rate used to adjust the speed of gradient descent. After a specific number of iterations, $J_{sparse}(W,b)$ is small enough to demonstrate $\hat{x} = x$, $h^{(2)}$ is the feature of the input and $W^{(2)}$ is the feature extractor, which is then set as a kernel function for CNN. Moreover, if there are many convolution layers, the same feature maps of different layers share the same kernel function while the kernel functions of different feature maps within the same layer are different. That is why CNN can recognize the object in different locations of a picture.

The convolutional features are then passed to the activation layer to activate the feature values. One of the most common activation functions is Rectified Linear Unit (ReLU), which ensures the signals that are strongly associated with past references propagate more efficiently for identification. The activated convolutional features are passed to subsampling (pooling) layer. In this layer, the convolutional features can be "smooth" or reduced in size. The common practice is to use the statistical features (such as maximum and average) of adjacent elements of the matrix within the pooled size range as the pooling feature. Therefore, the pooling operation brings about a certain degree of translation invariance. And by introducing a rotating training data set, it can also obtain a certain degree of rotation invariance. When the convolution operation is to be connected to another layer of operation, it needs to pass through the full connectedness layer, where neurons of previous layers are connected to every neuron in subsequent layers. It means all possible reasoning pathways from the input to the output are considered.

3 Classification

The softmax regression method is used for classification, which can classify data into multiple independent categories. This method is a generalisation of logistic regression, which can only classify data into two categories. *K* logistic regressions can also classify data into *K* categories, while these categories may overlap. The softmax regression method is shown as follows. Given the training set $\left\{\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)\right\}$ of *m* labelled examples, where $x^{(i)} \in \Re^n$ is an input feature, and $y^{(i)} \in \{1, \ldots, k\}$ is the coresponding category. The probability estimation of the class label from k different categories is:

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

where $\theta_1, \theta_2, \ldots, \theta_k \in \Re^{n+1}$ is the model parameters need to be optimised, and $h_\theta(x^{(i)})$ is normalised by dividing $\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$. The optimised parameters for the whole m inputs are obtained by solving the minimum cost function:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] + \frac{\lambda_s}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2 \tag{10}$$

where $1\{\bullet\}$ is the indicator function, which means if the statement in the curly braces is true, then $1\{\bullet\} = 1$, otherwise $1\{\bullet\} = 0$. λ_s is a weight decay parameter, and $\frac{\lambda_s}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2$ is the weight decay term used to penalise large parameter values. An iterative optimisation algorithm is used to obtain the minimum of $J(\theta)$, where the gradient is:

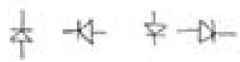
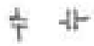
$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\begin{pmatrix} 1\{y^{(i)} = j\} \\ -p\left(y^{(i)} = j \mid x^{(i)}; \theta\right) \end{pmatrix}\right] + \lambda\theta_j \tag{11}$$

It can then use the standard optimisation package to obtain the optimal parameters for classification.

4 Data set and training process

During the training process, unsupervised feature learning and supervised training of classification methods are performed, so images and their labels are required. The data set built to train the recognition network contains three types of electronic components: capacitors, resistors, and diodes. Each component has 64 images drawn by different people, so there are 192 images in total. Originally, hand-drawn electronic components on A4 paper were scanned into a computer in black and white using a scanner, and each image was reduced in size and standardised by the method shown in Abhijeet and Chachan (2007). The entire picture was cut into small pieces, and then a small block of the picture containing the component was further reduced using a sliding window of the desired size to form a picture of 28×28 small pixel size. After all the pictures were picked out, 1/4 pictures were rotated 90 degrees, 1/4 pictures were rotated 180 degrees, 1/4 pictures were rotated 270 degrees and the others remained unchanged to make it more resemble the actual situation. From these three types of component images, 150 images were randomly selected with the same probability as the training data set, and the remaining 42 were used as the test data set for testing the recognition accuracy. Therefore, components contained in the data set were shown in Table 1.

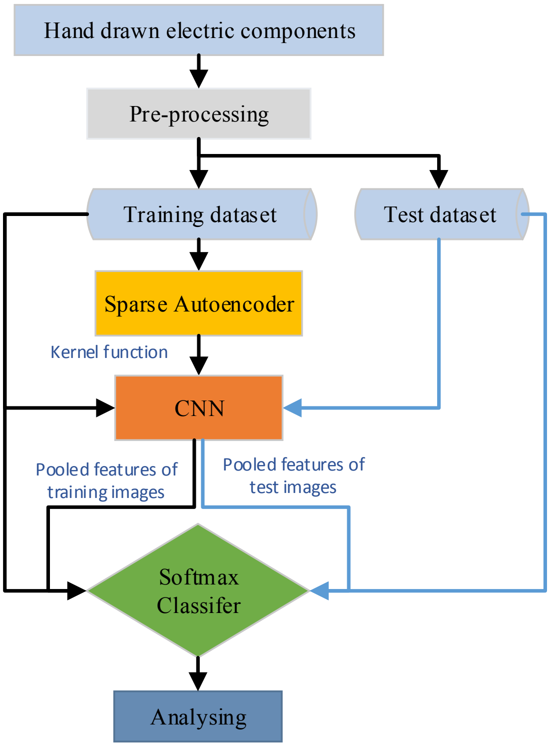
Table 1 Training data set examples

Components	*Hand-drawn images*
Diodes	
Resistors	
Capacitors	

5 Recognition process

The CNN used herein consists of a convolution layer, an activation layer and a pooling layer. All full-size images in the training data set are convolved one by one with the kernel function to obtain the convolutional features in the convolution layer and then the convolutional features are activated. The average pooling method is used in pooling layer, to reduce the size of the convolutional features and obtain pooling features. The softmax classifier is then trained by a supervised training method based on the pooling features and their corresponding labels. Finally, the test data set is convolved and pooled to obtain pooling features by using the trained CNN and classified by the softmax classifier to verify the accuracy of the recognition process. The process is shown Figure 2.

Figure 2 Flowchart of the recognition process

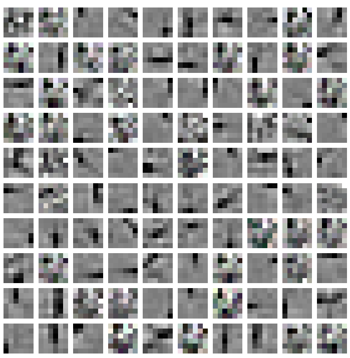


6 Experimental result and analysis

In this section, the network is trained and tested with the data set obtained in Section 4. The confusion matrix is used as an indicator of evaluation. The confusion matrix is an effective and simple way to judge the advantages of multiple classifications. It not only displays the correct proportion of the classification results but also shows the error location when the classification is wrong.

This work was implemented in MATLAB R2016a. During the training process operation, 1500 small patches with 6 × 6 pixels randomly selected from the training data set were used for training the sparse autoencoder and obtained a kernel function. Parameters of the sparse autoencoder were $\lambda = 0.003$, $\beta = 5$, and $\rho = 0.035$. Train for 500 iterations, extract 100 features, and then use their corresponding weights as the kernel function in CNN. The features are shown in Figure 3.

Figure 3 The extracted features of the sparse autoencoder



The training data set formed pooling features in the CNN using the kernel function for training the softmax classifier. In softmax classifier, $\lambda_s = 0.0001$, and the number of iterations was set to 200. If the cost function result is small enough, the iteration will terminate early, so the iteration number will be less than 200.

In the test operation, the test data set was also convoluted by the same kernel function through the CNN and then fed to the classifier. The classification result is shown in Figure 4, where each element is represented by its capital initial of the name for simplicity. Test result has shown that the method can identify hand-drawn electronic components in different directions. Moreover, the confusion matrix is shown in Figure 5. The confusion matrix showed the correct ratio of 95.2%, and two images of capacitors were misclassified to diodes while the others were classified correctly.

Figure 4 The test result

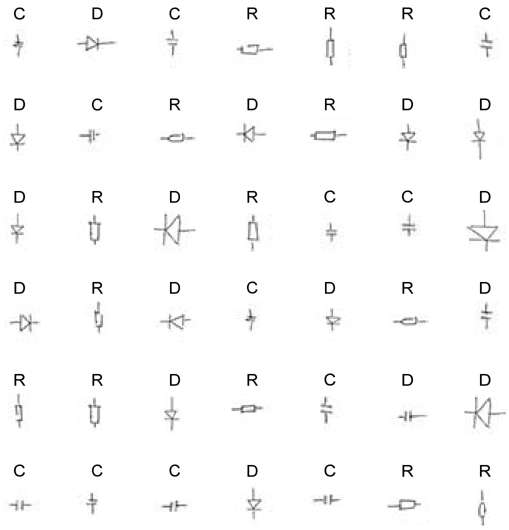
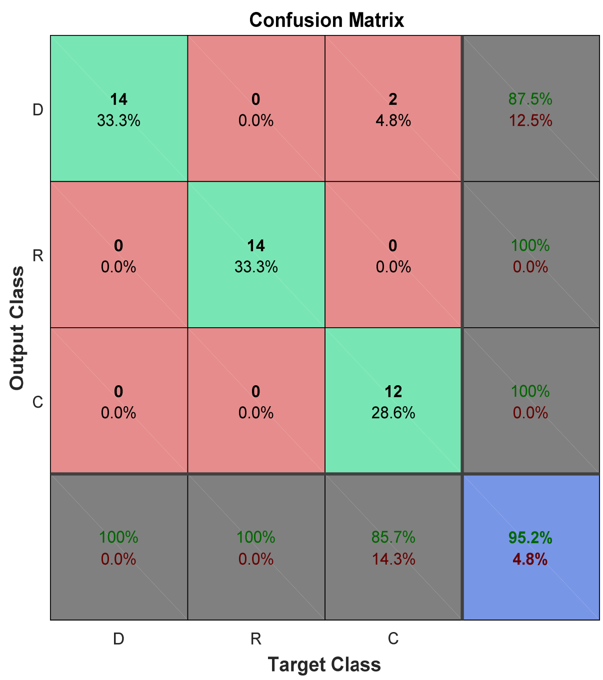


Figure 5 The confusion matrix



7 Conclusion

The hand-drawn electronic component recognition method using CNN and softmax classifier is presented in this paper. The features of electronic components are extracted by CNN and the components are finally classified into different categories by the trained softmax classifier. This method has not only a certain degree of rotation invariance but also a high identification accuracy which is over 95%. This method brings a new thought to the hand-drawn electronic components recognition. The future work will focus on model training with more electronic components and developing abundant hand-drawn circuit recognition method based on the proposed component recognition method.

Acknowledgement

This work is supported by the National Key R&D Program under Grant of China (Grant No. 2017YFF0211400) and Key R&D Program of Jiangsu Province, China (Grant No. BE2018370).

References

Abhijeet, A.K. and Chachan, P.K. (2007) *Recognition of Electrical and Electronics Components*, Doctoral dissertation, National institute of technology, Rourkela, India.

Angadi, M. and Naika, R.L. (2014) 'Handwritten circuit schematic detection and simulation using computer vision approach', *International Journal of Computer Science and Mobile Computing*, Vol. 3, No. 6, pp.754–761.

Chen, H., Lin, P., Emrith, K., Narayan, P. and Yao, Y. (2017) 'Ensemble-empirical-mode-decomposition based micro-Doppler signal separation and classification', *International Journal of Computer Applications in Technology*, Vol. 56, No. 4, pp.253–263.

Deng, W., Zhang, H., Li, Y. and Gao, F. (2018) 'Research on target recognition and path planning for EOD robot', *International Journal of Computer Applications in Technology*, Vol. 57, No. 4, pp.325–333.

Di, W., Wang, M., Sun, X., Kang, F., Xing, H., Zheng, H. and Bian, J. (2018) 'Identification of rock bolt quality based on improved probabilistic neural network', *International Journal of Modelling, Identification and Control*, Vol. 30, No. 2, pp.105–117.

Edwards, B. and Chandran, V. (2000) 'Machine recognition of hand-drawn circuit diagrams', *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal*, Istanbul, Turkey, Vol. 6, pp.3618–3621.

Gennari, L., Kara, L.B., Stahovich, T.F. and Shimada, K. (2005) 'Combining geometry and domain knowledge to interpret hand-drawn diagrams', *Computers and Graphics*, Vol. 29, No. 4, pp.547–562.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. (2016) *Deep Learning*, Vol. 1, Cambridge, Massachusetts.

Guo, Q., Wang, F., Lei, J., Tu, D. and Li, G. (2016) 'Convolutional feature learning and hybrid CNN-HMM for scene number recognition', *Neurocomputing*, Vol. 184, pp.78–90.

Hamdad, N. and Hammouche, K. (2018) 'Hilbert Huang transform and pattern recognition to detect defects in induction motor', *International Journal of Modelling, Identification and Control*, Vol. 29, No. 4, pp.352–358.

He, R., Wu, X., Sun, Z. and Tan, T. (2018) 'Wasserstein CNN: learning invariant features for NIR-VIS face recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 41, No. 7, pp.1761–1773.

Khuwaja, G.A. (2003) 'LVQ base models for recognition of human faces', *International Journal of Computer Applications in Technology*, Vol. 16, No. 4, pp.181–193.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, Vol. 521, No. 7553, pp.436–444.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, Vol. 86, No. 11, pp.2278–2324.

Liu, Y. and Xiao, Y. (2013) *Circuit Sketch Recognition*, Department of Electrical Engineering, Stanford University, Stanford, CA.

Ng, A., Ngiam, J., Foo, C.Y., Mai, Y. and Suen, C. (2010) *UFLDL tutorial*. Available online at: http://deeplearning.stanford.edu/wiki/index. php/UFLDL_Tutorial.

Patare, M.D. and Joshi, M.S. (2016) 'Hand-drawn digital logic circuit component recognition using SVM', *International Journal of Computer Applications*, Vol. 143, No. 3, pp.24–28.

Rabbani, M., Khoshkangini, R., Nagendraswamy, H.S. and Conti, M. (2016) 'Hand drawn optical circuit recognition', *Proceeding of the Seventh International Conference on Intelligent Human Computer Interaction*, Allahabad, India, Vol. 84, pp.41–48.

Sharafian, A. and Ghasemi, R. (2017) 'Stable state dependent Riccati equation neural observer for a class of nonlinear systems', *International Journal of Modelling, Identification and Control*, Vol. 28, No. 3, pp.256–270.

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R.M. (2016) 'Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning', *IEEE Transactions on Medical Imaging*, Vol. 35, No.5, pp.1285–1298.

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. (2015) 'Multi-view convolutional neural networks for 3d shape recognition', *Proceedings of the IEEE International Conference on Computer Vision*, Washington, DC, USA, pp.945–953.

Tao, Q.Q., Zhan, S., Li, X.H. and Kurihara, T. (2016) 'Robust face detection using local CNN and SVM based on kernel combination', *Neurocomputing*, Vol. 211, pp.98–105.

Vedaldi, A. and Lenc, K. (2015) 'Matconvnet: convolutional neural networks for MATLAB', *Proceedings of the 23rd ACM international conference on Multimedia*, New York, USA, pp.689–692.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K.J. (1990) 'Phoneme recognition using time-delay neural networks', *Readings in Speech Recognition*, The Elsevier Press, Amsterdam, pp.393–404.

Wang, G.Y, Li, Z.M. and Wang, J.S. (2001) 'The recognition system of handwriting electronic component symbols based on neural network', *Journal of Jiangsu University of Science and Technology*, Vol. 22, No. 2, pp.82–86.

Wang, Z.Z., Xu, X.X., Li, L. and Liu, W. (2018) 'Research on face recognition based on DRNLGBP', *International Journal of Computer Applications in Technology*, Vol. 57, No. 2, pp.104–111.

Yao, L. and Wang, H. (2018) 'Fault diagnosis and fault tolerant control for the non-Gaussian nonlinear stochastic distribution control system using Takagi-Sugeno fuzzy model', *International Journal of Modelling, Identification and Control*, Vol. 29, No. 1, pp.22–30.