

Object Recognition System of Sonar Image Based on Multiple Invariant Moments and BP Neural Network

Jianning Han¹, Peng Yang¹ and Lu Zhang¹

School of Information and Communication Engineering, North University of China, Taiyuan Shanxi China, 030051

¹*E-mail: super_lens_hjn@foxmail.com*

Abstract

Object Recognition System of Sonar Image plays an important role in the field of underwater defense. According to pattern recognition theory, principle processes of a typical object recognition system is introduced. To achieve robust performance, image de-noising, sharpening, binary transformation, edge detection and other image processing techniques are discussed. The paper presents a novel object recognition system using multiple invariant moments as the main feature of the object, and the detected feature is trained by BP neural network so that the classification error can be minimized. Finally, we implemented the proposed approach by visual air plane recognition, the experimental results demonstrates the robustness and efficiency of the presented approach.

Keywords: *Sonar image, Target recognition, Feature extraction, Image edge detection, Multiple Invariant Moments, BP Neural Network*

1. Introduction

Today, the sonar image is one of the most important information source in underwater defense system. The key technology of sonar image processing is to realize the underwater automatic target recognition. Digital image processing and machine vision play more the more important role in the automation field research [1]. Based on pattern recognition and digital image processing technology, this paper discusses the techniques of image denosing, feature extraction, pattern classification and object identification. Hu's moment theory is introduced to represent the image, and characters of processed images are employed to train BP neural network [2]. Finally, it is convinced that the proposed method could classify and recognize the target image robustly and efficiently.

1.1. Page Setup: Margins and Layout

Pattern recognition aims to classify data (patterns) based either on a priori knowledge or on statistical information extracted from the patterns. It's playing a important role in information technology and artificial intelligence, and usually used in image processing and analysis, speech recognition, acoustics classification, communication, diagnosis by computer, and data-mining[3].

Principle processes of an image recognition system are depicted below in Figure 1. Because most of the test objects are non-electronical parameters, it is first step to convert them to electronic forms. After that A/D transformation from the analog signal to digital, normally this step can be done by PCs or digital signal processors (DSP).

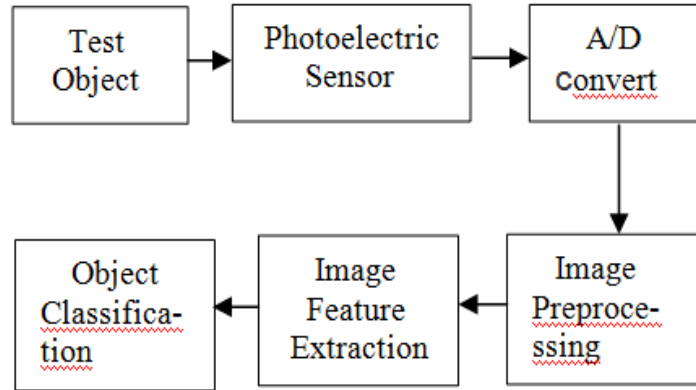


Figure 1. Flow Chart of a Typical Image Recognition System

During the digitalization procession some noise will be introduced due to the quantization error, it is necessary to improve the noisy image to a better image. The above processes are called preprocessing, which can segment the target object from the input image [4].

Feature extraction is intended to extract a set of features to represent the abstract information of the input image after preprocessing, and the detected features can be represented mathematically in terms of vector. The characteristic vector is a sparse representation of the visual target object, which captures main information of the image, and also provides enough information for classification [5].

1.2. Preprocesses of Images and Features Extraction Based on Multiple Invariant Moment

A sonar image is stored in PCs as a matrix. In order to stabilize the feature extraction operation and good recognition to the image preprocessing, some transformation, namely preprocesses are adopted. After preprocesses computer can get a more clear and easy-operation digital image [6].

Because the information condensed in the raw data is too much and redundant for target image recognition if we use the information directly which may have low recognition efficiency, thus a series methods about features selection and extraction are introduced. Features selection and extraction is one of the most crucial steps in image recognition, which will effect on coming parts of image recognition [7].

Features extraction is a kind of image transformations, which can transfer high-dimension features vector to low-dimension feature. The feature extraction achieves dimensional compression, while maintain the necessary information, which is more representative and essential to the image [8].

Shape description is an important feature of the target images. It is found that the invariants moment has good effect in classification for the recognition system. Invariant moments is a statistical mathematical character of the image, however the image is translated, rotated or scaled. In 1962 M.K.Hu first proposed continuous function definition of expression of the invariant moment, and then it had been widely applied in image recognition field. On the basis of Hu's theory, Y. R. Wong defined algorithm of all moments in discrete conditions [9].

Suppose a image as a function, $f(x, y)$, which is a piecewise function in limited field. Thus moments of $f(x, y)$ can be defined as follow:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \quad (1)$$

m_{pq} is usually named geometry moment or regular moment and its order n is defined as: $n = p + q$. There is a one-to-one Correspondence between $\{m_{pq}\}$ and the image $f(x, y)$.

And the central moments of an image are given as:

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_0)^p (y - y_0)^q f(x, y) dx dy \quad (2)$$

with x_0 and y_0 are barycenter of the image, and $x_0 = m_{10}/m_{00}$, $y_0 = m_{01}/m_{00}$ respectively.

For a digital image $f(x, y)$ with size of $M \times N$, its moments and central moment can be defined as follow:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) dx dy \quad (3)$$

$$\mu_{pq} = \sum_x \sum_y (x - x_0)^p (y - y_0)^q f(x, y) dx dy \quad (4)$$

According to the definition above moments, whose order is less than three, are defined as:

$$\mu_{00} = m_{00} \quad (5)$$

$$\mu_{10} = \mu_{01} = 0 \quad (6)$$

$$\mu_{11} = m_{11} - x_0 m_{10} \quad (7)$$

$$\mu_{20} = m_{20} - x_0^2 m_{00} \quad (8)$$

$$\mu_{02} = m_{02} - y_0^2 m_{00} \quad (9)$$

$$\mu_{12} = m_{12} - x_0 m_{02} - 2y_0 m_{11} + 2y_0^2 m_{10} \quad (10)$$

$$\mu_{21} = m_{21} - y_0 m_{20} - 2x_0 m_{11} + 2x_0^2 m_{01} \quad (11)$$

$$\mu_{30} = m_{30} - 3x_0 m_{20} + 2x_0^3 m_{00} \quad (12)$$

$$\mu_{03} = m_{03} - 3y_0 m_{02} + 2y_0^3 m_{00} \quad (13)$$

For a gray image, these regulations can be found:

(1) μ_{20} and μ_{02} are inertia moments which center on gray gravity vertical and horizontal respectively. If $\mu_{20} > \mu_{02}$, image is stretched in horizontal direction, otherwise image is stretched in vertical direction.

(2) Amplitudes of μ_{30} and μ_{03} are characters which can be used to measured the symmetry to vertical and horizontal axes. If $\mu_{30} = 0$, the image is axisymmetric to the vertical axis, and if $\mu_{03} = 0$, the image is axisymmetric to the horizontal axis. To summarize information above, μ_{pq} describe the relationship between regional characters and image gravity.

The equations above are sensitive to rotation and scale transformation; it is a good choice to adopt normalization to overcome these adverse effects. Definitions of normalization are defined as:

$$\eta_{pq} = \mu_{pq} / \mu_{pq}^r$$

$$\text{with } r = \frac{p+q}{2} + 1, (p+q = 2, 3, \dots) \quad (14)$$

Seven characters, which can maintain the information of an image after translation, scale, and rotation, can be derived from normalized second-order moments and third-order moments.

$$\phi_1 = \eta_{20} + \eta_{02} \quad (15)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (16)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (17)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \quad (18)$$

$$\phi_5 = (\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - (3\eta_{21} + \eta_{03})^2 \right] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \quad (19)$$

$$\phi_6 = (\eta_{20} - \eta_{02})^2 \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\ + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (20)$$

$$\phi_7 = (3\eta_{21} - 3\eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] \\ + (3\eta_{12} - 3\eta_{30})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \quad (21)$$

1.3. BP neural Network

BP (Back Propagation) neural network is now widespread application of neural network model, which is a multilayer forward feed network using error back propagation algorithm for training. BP network can learn and storage of input-output model mapping relation, without the feeding with the mapping relationship of mathematical equations in advance. It use the rules of learning the steepest descent method, through propagation to adjust network weights and threshold constantly, make the most minimum square sum of network's error [10].

Training neural network is essentially a non-linear least squares problem, and thus can be solved by a class of non-linear least squares algorithms. Among them, the Levenberg-Marquardt is a trust region based method with hyper-spherical trust region. This method work extremely well in practice, and is considered the most efficient algorithm for training median sized artificial neural networks. The Levenberg-Marquardt algorithm [L44, M63], which was independently developed by Kenneth Levenberg and Donald Marquardt, provides a numerical solution to the problem of minimizing a nonlinear function. It is fast and has stable convergence. In the artificial neural-networks field, this algorithm is suitable for training small- and medium-sized problems [11].

The Levenberg-Marquardt algorithm blends the steepest descent method and the Gauss-Newton algorithm. Fortunately, it inherits the speed advantage of the Gauss-Newton algorithm and the stability of the steepest descent method. It's more robust than the Gauss-Newton algorithm, because in many cases it can converge well even if the error surface is much more complex than the quadratic situation. Although the

Levenberg–Marquardt algorithm tends to be a bit slower than Gauss–Newton algorithm (in convergent situation), it converges much faster than the steepest descent method. The Levenberg–Marquardt algorithm was designed to approach second order training speed with out having to compute Hessian matrix. When the performance function has the form of a sum of squares that is

$$F(\underline{w}) = \frac{1}{2} e^T e = \frac{1}{2} e^T (w) e(w) = \frac{1}{2} \sum_i^k \sum_j^p (O_{ij} - t_{ij})^2$$

where $\underline{W} = [w_1, w_2, w_3, \dots, w_N]^T$ consists of all weights of the network, the function of sum of squared errors is defined as

$$F(W) = \frac{1}{2} e^T e$$

In order to simplify the calculating process, Jacobian matrix J is introduced as

$$J = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{1,M}}{\partial w_1} & \frac{\partial e_{1,M}}{\partial w_2} & \dots & \frac{\partial e_{1,M}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p,1}}{\partial w_1} & \frac{\partial e_{p,1}}{\partial w_2} & \dots & \frac{\partial e_{p,1}}{\partial w_N} \\ \frac{\partial e_{p,2}}{\partial w_1} & \frac{\partial e_{p,2}}{\partial w_2} & \dots & \frac{\partial e_{p,2}}{\partial w_N} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{p,M}}{\partial w_1} & \frac{\partial e_{p,M}}{\partial w_2} & \dots & \frac{\partial e_{p,M}}{\partial w_N} \end{bmatrix}$$

Newton's method for minimizing objective function is generated using well known recurrence formula.

$$W_{i+1} = W_i - H^{-1} \nabla F(W) \tag{22}$$

When $F(W) = \frac{1}{2} e^T e$ and $\nabla F(W)$ is the gradient of $F(W)$, then the Hessian matrix can be approximated as $H \approx J^T J$, and the gradient can be computed as

$$g = J^T e$$

where the Jacobian matrix J contains the first derivatives of the network errors with respect to weights and biases, and e is a vector of network errors. The Gauss-Newton update formula can be

$$W_{i+1} = W_i - (J_i^T J_i)^{-1} J_i^T \varepsilon_i \tag{23}$$

In order to make sure that the approximated Hessian matrix $J^T J$ is invertible,[12] Levenberg–Marquardt algorithm introduces another approximation to Hessian matrix:

$$H \approx J^T J + \mu I \tag{24}$$

where μ is always positive, called combination coefficient. It is the identity matrix. From Equation 24, one may notice that the elements on the main diagonal of the approximated Hessian matrix will be larger than zero. Therefore, with this approximation (Equation 24), it can be sure that matrix H is always invertible. By combining Equations 23 and 24, the update rule of Levenberg–Marquardt algorithm can be presented as

$$W_{i+1} = W_i - (J_i^T J_i + \lambda I)^{-1} J_i^T \varepsilon_i \quad (25)$$

2. Experimental Results

It is applied VC++6.0 to realize image denoising, smoothing, sharpening, edge extraction, and binaryzation, the processes are illustrated as follows:

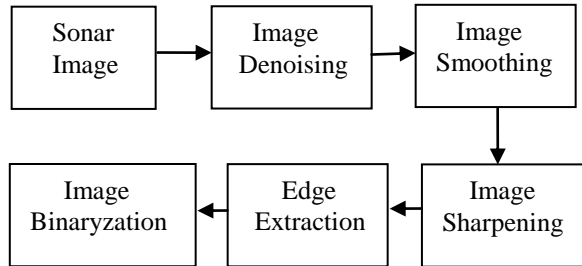


Figure 2. Image Preprocessing Flowchart

Comparing the images preprocessed to original, it is easy to see that edge of the pre-processed image is clearer and background is much less significant.

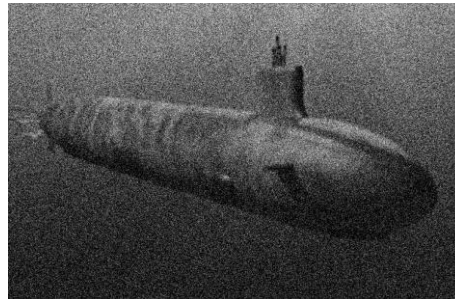


Figure 3. Sonar Image



Figure 4. The Pre-processed Image

In order to compare the experimental results, the edge of the top and the whole submarine are extracted. Flowchart of calculating seven invariant moments

Edge extraction outcome in condition that is mentioned above is shown as flow.

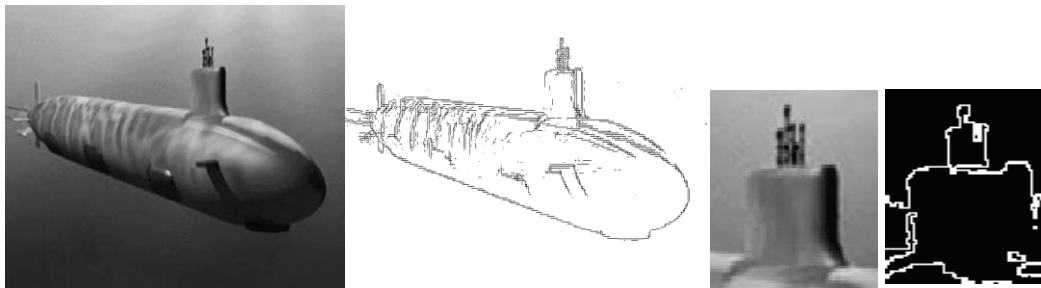


Figure 6. Result of Edge Extraction

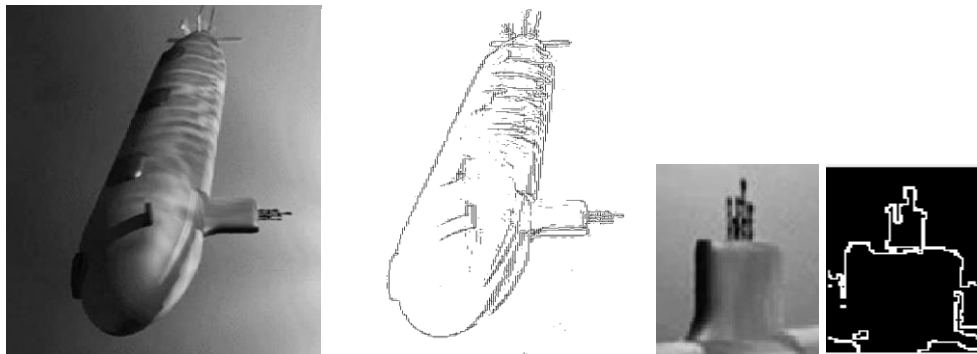


Figure 7. Image After Rotating 90°

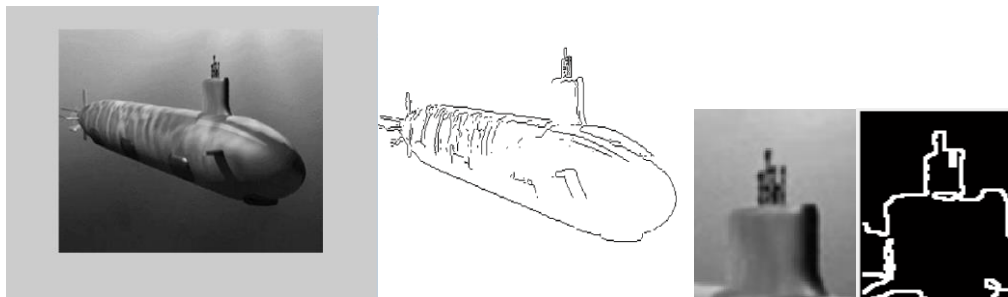


Figure 8. Image After Resizing in Half

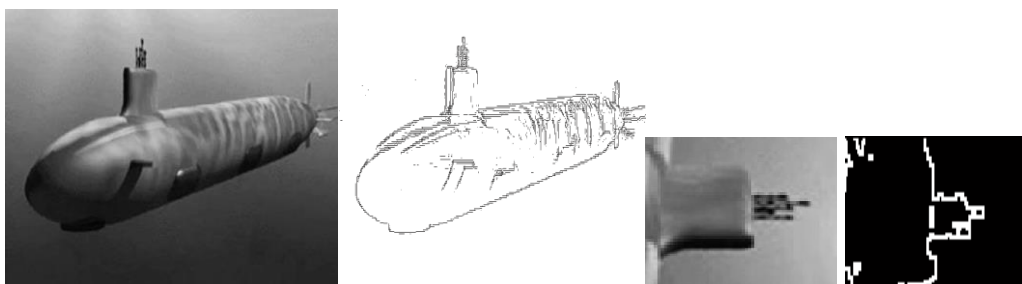


Figure 9. Image After Mirroring

Using the above edge information, the invariant moment of image is calculated according to the process in the Figure10. Table 1 shows the invariant moment from the whole edge of submarine. Table 2 shows the invariant moment from the top edge of

the submarine. By analyzing seven invariant moment of image original, rotated 90°, sized into half and mirrored, we can find that the outcome is almost constant.

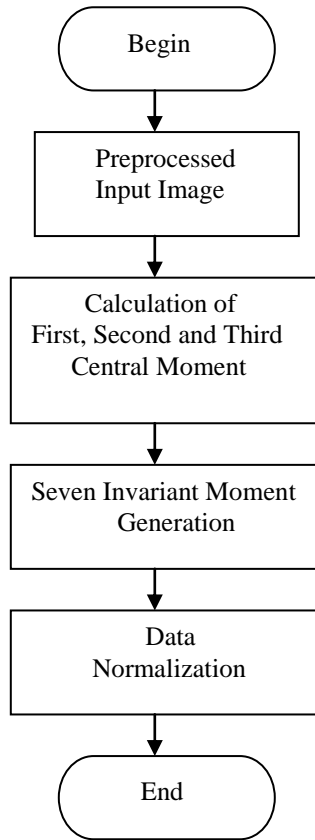


Figure 10. Flaw Chart of Seven Invariant Moments Computation

Table 1. The Moments by the Whole Edge of Submarine

Invariant Moments	Original Image	Image after zooming in half	Image after rotating 90°	Image after mirroring
ϕ_1	0.6075e-004	0.6109e-004	0.6075e-004	0.6075e-004
ϕ_2	0.5018e-009	0.5266e-009	0.5018e-009	0.5018e-009
ϕ_3	0.3130e-015	0.3116e-015	0.3085e-015	0.3085e-015
ϕ_4	0.1085e-014	0.2176e-014	0.1805e-014	0.1805e-014
ϕ_5	-0.4446e-029	-0.1618e-028	-0.4446e-029	-0.4446e-029
ϕ_6	-0.6149e-019	-0.7267e-019	-0.6149e-019	-0.149e-019
ϕ_7	-0.6332e-029	-0.6975e-029	-0.6332e-029	-0.6332e-029

Table 2. The Moments by the Top Edge of the Submarine

Invariant Moments	Original Image	Image after zooming in half	Image after rotating 90°	Image after mirroring
ϕ_1	2.0535e-003	2.0650e-003	2.0535e-003	2.0535e-003
ϕ_2	6.6961e-008	1.7801e-008	1.6961e-008	1.6961e-008
ϕ_3	1.0582e-014	1.0539e-014	1.0582e-014	1.0582e-014
ϕ_4	6.1040e-014	7.3569e-014	6.1040e-014	6.1040e-014
ϕ_5	-1.5023e-028	-5.4693e-028	-1.5023e-028	-1.5023e-028
ϕ_6	-2.0786e-018	-2.4563e-018	-2.0786e-018	-2.0786e-018
ϕ_7	-2.1406e-028	-2.3576e-028	-2.1406e-028	-2.1406e-028

This paper use seven invariant moments as input of neural network, apply Levenber-Maquardt algorithm for training; and the expectation of the target object of is set to 11. The system is programmed to learning and recognition in environment of VC++. After repeated experiments, the design of network parameter is as follow:

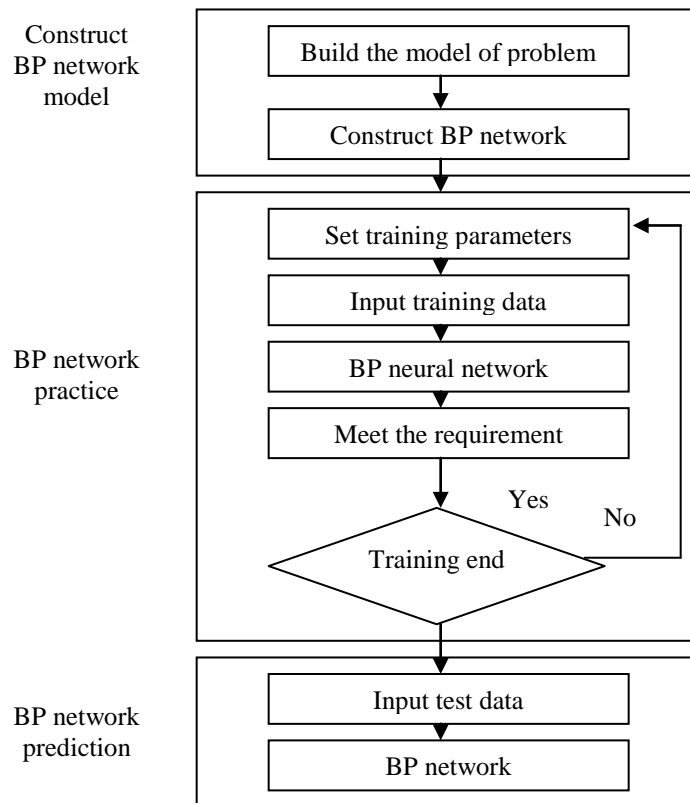


Figure 11. BP Network Model to Realize Flow Chart

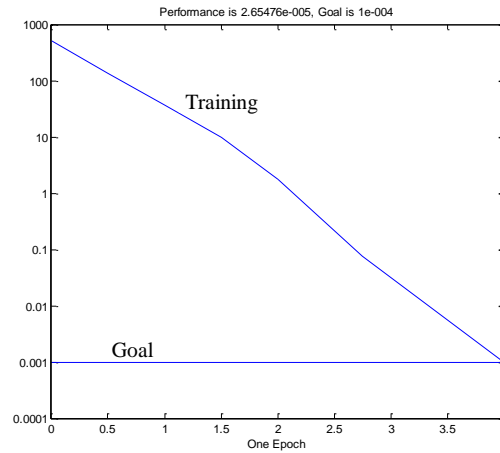


Figure 12. The Steps of Training by the Whole Edge of Submarine

Number of input unit is 7, number of output unit is 1, number of hidden layer is 3 and layer number of the network is 3. According to the flow chart in Figure 11, the system adopts training function of Levenber-Maquardt as learning algorithm, and steps of training are depicted by the following picture. we get the following conclusion: in the successful design of BP neural network model based on a reasonable sample of data, neural networks through the training sample, learning to intrinsic regularity of the sample relating to the environment, which can also be input for the future give the correct response, achieving a successful teaching quality assessment.

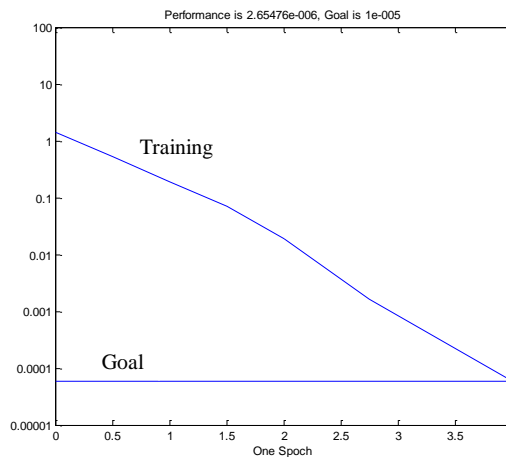


Figure 13. The Steps of Training by the Top Edge of the Submarine

3. Conclusions

We designed a system for sonar image object recognition based on multiple invariant moment theory, study the invariant moments in the variant situations of rotation, scale and viewpoint. We approach BP neural network as recognition rule for its easy implementation, and high efficient training process. Experimental results show sonar image recognition system based on theory of invariant moments and the BP neural network can achieve accurate results as well as robustness to environment changes.

Acknowledgements

This project was sponsored in part by The National Natural Science Foundation of Chinese NO.61302159.

References

- [1] J. P. MacSleyne, J. P. Simmons and M. De Graef, "On the use of 2-D moment invariants for the automated classification of particle shapes", *Acta Materialia*, vol. 56, no. 3, (2008), pp. 427-437.
- [2] L. Xueyan, G. Shuxu, G. Fengli, *et al.*, "Vein pattern recognitions by moment invariants *Bioinformatics and Biomedical Engineering*", 2007. ICBBE (2007), the 1st International Conference on. IEEE, (2007), pp. 612-615.
- [3] M. F. Moens, "Information extraction: algorithms and prospects in a retrieval context", Springer, (2006).
- [4] R. Oshana, "DSP software development techniques for embedded and real-time systems", Newnes, (2006).
- [5] C. H. Wei, Y. Li, W. Y. Chau, *et al.*, "Trademark image retrieval using synthetic features for describing global shape and interior structure", *Pattern Recognition*, vol. 42, no. 3, (2009), pp. 386-394.
- [6] T. A. C. Zitter, C. Huguen and J. M. Woodside, "Geology of mud volcanoes in the eastern Mediterranean from combined sidescan sonar and submersible survey", *Deep Sea Research Part I: Oceanographic Research Papers*, vol. 52, no. 3, (2005), pp. 457-475.
- [7] V. Sugumaran, V. Muralidharan and K. I. Ramachandran, "Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing", *Mechanical Systems and Signal Processing*, vol. 21, no. 2, (2007), pp. 930-942.
- [8] Y. Zilu, L. Jingwen and Z. Youwei, "Facial expression recognition based on two dimensional feature extraction *Signal Processing*", ICSP 9th International Conference on IEEE, (2008), pp. 1440-1444.
- [9] N. A. Bakar and S. M. Shamsuddin, "United zernike invariants for character images", *Visual Informatics: Bridging Research and Practice*. Springer Berlin Heidelberg, (2009), pp. 498-509.
- [10] K. L. Zhou and Y. H. Kang, "Neural Network Mold and Design in MATLAB", Tsinghua University Press, Beijing, (2007).
- [11] A. M. Aqlan, W. F. A. El-Wahed and M. A. El-Wahed, "A Hybrid Extreme Learning Machine with Levenberg-Marquardt Algorithm using AHP Method", *INFOS*, (2008), pp. 110-117.
- [12] A. Ranganathan, "The levenberg-marquardt algorithm", *Tutorial on LM Algorithm*, (2004).

