

A Real-time Small Immobile Object Recognition System Using Wavelet Moment Invariants

Jiaojiao Gu, Zhao Wang, Haitao Song, Han Xiao, Wenhao He and Kui Yuan

Institute of Automation, Chinese Academy of Sciences

95 Zhongguancun East Road, Beijing, China

gujiaojiao2012@ia.ac.cn

Abstract - In this paper, a real-time small immobile object recognition system is implemented using wavelet moment-based back-propagation(BP) neural network classifier. The system is composed of a camera and an image acquiring and processing board developed by our research team. An FPGA chip and a DSP chip are embedded in the image board as the major calculation units, which make real-time computation possible. First, wavelet moment invariants of training samples are integrated with BP neural network to construct the classifier on the host computer. Then, real-time object detection and classification experiments are conducted according to the classifier on the image acquiring and processing board. Experiment results show that the algorithm can detect and classify different small immobile object types efficiently.

Index Terms - small immobile object, real-time recognition, wavelet moment invariants, embedded system, BP neural network.

I. INTRODUCTION

On modern high-tech battlefield, the distance between the military camera and hostile object is usually very far away, which results in the object occupying quite a low percentage in the image [1]. This restriction makes it almost impossible to extract any gray degree features or texture features from the small object. Detecting and recognizing of small moving targets is relatively easy, however, as the battlefield environment is wretched, when the small threatening target remains motionless, to detect and recognize it becomes more difficult. What's more, military threatening objects such as missiles and fighter planes proceed at high speed, once they stop remaining stationary and suddenly launch offensives, severe damages will be caused in a few seconds. Therefore, it is particularly significant to detect and recognize the immobile threatening small targets. In recent years, many researchers have proposed lots of methods to solve the above problems, but most of them focused on the effectiveness and accuracy of the algorithm proposed in their papers[2-4]. However, to rapidly and accurately detect and identify the hostile objects is a more important factor that affects the success of modern high-tech war. So in this paper, we will not only pay attention to the precision, but also the speed.

General-purpose computer system is widely used to process the computer vision problems such as objects detecting and identifying. When handling algorithms with high computing complexity, the low processing speed of CPU is not able to meet the needs of real-time applications. Using an FPGA chip and a DSP chip as the main computing

elements, our research team developed an intelligent image acquiring and processing board to realize the real-time computation for complex algorithms[1].

Moment invariant refers to a global invariance with scaling, translation and rotation invariance. It is fairly time-consuming, susceptible to noise and merely applied in the situation where object images have obvious differences. Wavelet analysis is a kind of multi-scale analysis, and wavelet function is orthogonal or near orthogonal with low information redundancy. However, wavelet analysis is neither translation invariant nor rotation invariant. Even a slight change of the object image will cause great differences in wavelet feature. D.Shen and H.H.S Ip proposed wavelet moment invariant[5], which combines the object's local features with global features, and has strong robustness. Wavelet moment invariant has scaling, translation and rotation invariance. Thus, compared with Hu moments[6] and Zernike moments[7], it has better computation complexity and recognition effect in identifying different objects of similar shape.

Pattern recognition is a process of labeling the unknown objects based on knowledge already possessed. There are two main approaches to achieve this goal, the parametric and nonparametric approaches. The parametric approach, known as the statistical approach, requires a good assumption of the statistical distribution of the pattern data. On the other hand, the nonparametric approach, such as the neural network approach, does not require any statistical assumption of the pattern data. As a well known nonparametric approach, back-propagation(BP) neural network classifier is suited for a realization in hardware because of its regular structure[8].

The paper is organized as follows. Section II briefly introduces the intelligent image acquiring and processing embedded board developed by our research team. A general framework of this wavelet moment-based immobile small object recognition algorithm is outlined in Section III, which provides a detailed description of the methodologies. In Section IV, experiments are conducted to test the performance of the system and the algorithm. Section V makes conclusions of this paper.

II. INTELLIGENT IMAGE ACQUIRING AND PROCESSING EMBEDDED BOARD

An FPGA chip and a DSP chip are the core computing elements of the intelligent image acquiring and processing board (see Fig. 1). The FPGA used on the board is Altera

Cyclone III EP3C40F484, and the DSP on the board is TI's TMS320DM642. The board contains two SRAMs, two analog video input ports and one analog video output port. Analog video signal of PAL format is converted to BT.656 digital signal by the A/D converter TVP5150. Then the digital signal is transferred into the FPGA. The pins of the FPGA are connected with the two SRAMs, where the image data and results could be saved. Besides, the pins of DSP and FPGA are connected, which makes DSP convenient to access the memory resources in the FPGA and the SRAM on the board. The results and images can be output to the PC through the 100-Mbps Ethernet port. In addition, the gigabit Ethernet port can connect the high-speed digital camera to achieve faster image acquisition speed.

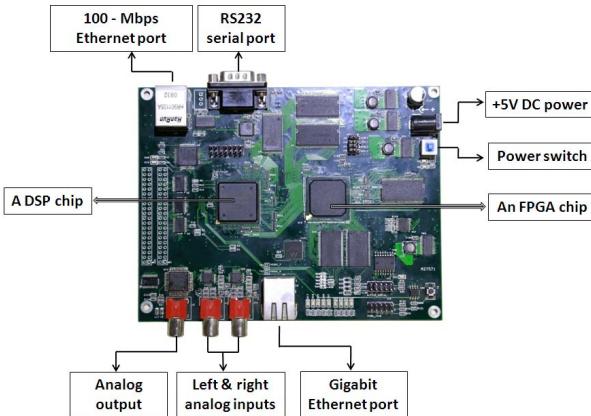


Fig. 1 The intelligent image acquiring and processing board.

III. WAVELET MOMENT-BASED IMMOBILE SMALL OBJECT RECOGNITION ALGORITHM

In this paper, the basic idea of the real-time immobile small object recognition system can be divided into four parts: taking 2-D images of objects from different angles with a CCD camera and dividing them into two parts (training image set and testing image set), extracting wavelet moment features of each image, choosing BP on 3-layer multilayer perceptron (MLP) neural network to train the training image set and obtain the classifier, identifying and labeling unknown objects according to the classifier. The overall implementation scheme is shown in Fig. 2.

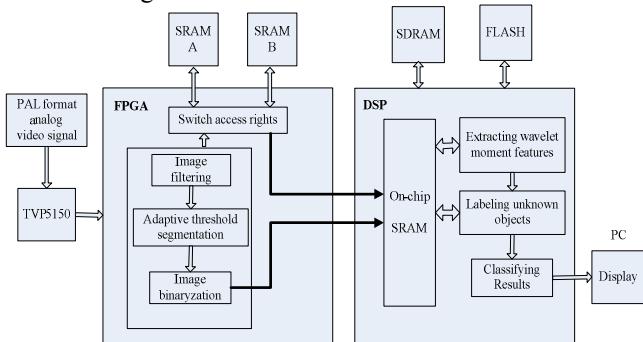


Fig. 2 The overall implementation scheme.

A. Image acquisition, segmentation and pre-processing

Fig. 3 shows the process of how to establish the image sets. The object is laid on an automatic rotating platform, and the camera is fixed in front of the platform. The camera keeps taking images of the object continuously while the turntable rotates. The acquired image's resolution is 720x576 pixels.

The algorithm is applied to two image sets, the first set consisting of three types of similar tanks and four types of similar cars (Fig. 4), including 825 training images and another 825 testing images. Column 1 shows original images objects of



Fig. 3 The process of acquiring object image set.

the seven kinds, while column 2 shows the corresponding binarized images through an adaptive threshold segmentation method. The second image set consisting of three kinds of objects—car, plane and man (Fig. 5), including 282 training images and another 123 testing images. Each category contains several seemingly similar but different objects. Column 1 shows part of the three kinds of objects, while column 2 shows the corresponding binarized images.

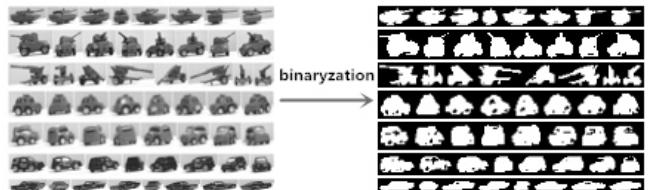


Fig. 4 A part of the 1st image set.

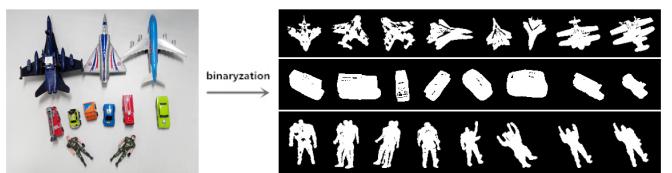


Fig. 5 A part of the 2nd image set.

B. Extracting discrete wavelet moment invariant features

In this paper, translation invariant and scaling invariant are achieved by using a normalization based on regular moments. Rotation invariant is achieved by extracting wavelet moment invariants.

1) Translation normalization

Let $f(x, y)$ represent a discrete 2-D binary image object in the (x, y) -coordinate. For an $M \times N$ image $f(i, j)$, its geometric moment invariants m_{pq} is

$$m_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j)$$

Where $p = 0, 1, 2, \dots$, $q = 0, 1, 2, \dots$.

Since the center of the shape is invariant to translation, rotation and scaling, the method of solving the translation problem is to locate the centroid of the shape. The coordinates of the center of the shape are

$$\bar{x} = m_{10} / m_{00}, \quad \bar{y} = m_{01} / m_{00}$$

Let $f1(i, j)$ represent a translation normalized 2-D binary image, it is of the same size as $f(i, j)$, i.e. $M \times N$. This way, we can obtain the translation normalized shape by changing the coordinates according to the following transformation:

$$\begin{aligned} \text{If } f(i, j) == 1, \text{ then } f1(i - \bar{y} + M/2, j - \bar{x} + N/2) &= 1 \\ \text{Else, } f1(i, j) &= 0 \end{aligned} \quad (1)$$

2) Scaling normalization

In Hu's moments, scaling invariant was obtained by normalizing the central moments by m_{00} . To simplify discussion, we extract the scaling factor based on Hu's idea. The scaling factor of the present object size, compared with the expected size, is

$$\alpha = \sqrt{m_{00} / \text{AREA}}$$

Where AREA is a constant, referring to the expected size of the object.

For $f1(i, j)$, let $f2(i, j)$ represent its scaling normalized 2-D binary image, its size is $(M/\alpha, N/\alpha)$. This way, we can obtain the scaling normalized shape by changing the coordinates according to the following transformation:

$$f2(i, j) = f1(i * \alpha, j * \alpha) \quad (2)$$

3) Image size adjusting

To make further processing easy, we adjust image $f2(i, j)$ into a new image $f3(i, j)$ of the unified size (Q, Q) . The object in $f3(i, j)$ is still in the image center and of the same size as in $f2(i, j)$.

$$\begin{aligned} \text{If } f2(i, j) == 1, \text{ then } f3(i + \frac{Q-M}{2}, j + \frac{Q-N}{2}) &= 1 \\ \text{Else } f3(i, j) &= 0 \end{aligned} \quad (3)$$

4) Polar coordinate transformation

Let $\text{Polar}(r, \theta)$ be the corresponding form of $f3(i, j)$ in the polar coordinate. The relationship between $f(x, y)$ and $\text{Polar}(r, \theta)$ is given as follows:

$$x = r * \cos(\theta), \quad y = r * \sin(\theta).$$

In this paper, we let $\Delta\theta = 2\pi/256$ and Q be 64, and suppose the origin of the polar coordinates is located in the image center. Then r is integer from 1 to $Q/2$. So the size of $\text{Polar}(r, t)$ is 32×256 .

$$\begin{aligned} \text{If } f3(r * \cos(t * \Delta\theta) + 32, r * \sin(t * \Delta\theta) + 32) &== 1, \text{ then} \\ \text{Polar}(r, t) &= 1; \\ \text{Else Polar}(r, t) &= 0 \end{aligned} \quad (4)$$

5) Extracting wavelet moment invariants

To get rotation invariant moments, typically the following generalized expression is used:

$$F_{pq} = \iint \text{Polar}(r, \theta) g_p(r) e^{jq\theta} r dr d\theta \quad (5)$$

Where F_{pq} is the pq -order moment, $g_p(r)$ is a function of radial variable r , and p and q are integer parameters.

In order to reduce the problem of feature extraction from a 2D image object to that from a 1D sequence, (5) is rewritten as follows:

$$F_{pq} = \int s_q(r) g_p(r) r dr \quad (6)$$

Where $s_q(r) = \int \text{Polar}(r, \theta) e^{jq\theta} d\theta$. Note that $s_q(r)$ is now a 1D sequence of variable r .

In this paper, we treat $\{g_p(r)\}$ in (6) as wavelet basis functions. Consider the family $\Psi_{a,b}(r) = 1/\sqrt{a} \psi((r-b)/a)$, where a ($a \in \mathbb{R}_+$) is a dilation parameter and b ($b \in \mathbb{R}$) is a shifting parameter. From now on, the basis functions $\{g_p(r)\}$ are replaced by wavelet basis functions $\{\Psi_{a,b}(r)\}$. We consider using the cubic B-spline wavelets which are optimally localized in space-frequency. The mother wavelet $\psi(r)$ of the cubic B-spline in Gaussian approximation form is

$$\psi(r) = \frac{4a^{n+1}}{\sqrt{2\pi(n+1)}} \sigma_w \cos(2\pi f_0(2r-1)) \cdot \exp\left(-\frac{(2r-1)^2}{2\sigma_w^2(n+1)}\right) \quad (7)$$

Where $n=3$, $a=0.697066$, $f_0=0.409177$ and $\sigma_w^2=0.561145$.

The values of parameters a and b are usually discrete. The discretization of the dilation parameter is done by choosing $a = a_0^m$, where m is an integer and $a_0 > 1$ or $a_0 < 1$. The discretization of the shifting parameter is done by choosing $b = nb_0 a_0^m$, where n is integer and $b_0 > 0$. Since the image size is always restricted in a domain $\{r \leq 1\}$, let parameter a_0 be set to 0.5 and parameter b_0 be set to 1, and the domains for m and n be restricted as follows:

$$\begin{cases} a = 0.5^m, & \text{where } m = 0, 1, 2, 3 \\ b = n \cdot 0.5^m, & \text{where } n = 0, 1, \dots, 2^{m+1} \end{cases}$$

Then the wavelet defined along a radial axis in any orientation is denoted by

$$\psi_{m,n}(r) = 2^{m/2} \psi(2^m r - n) \quad (8)$$

Now let us introduce a set of wavelet moment invariants for classifying objects, they are defined as follows:

$$\|F_{m,n,q}\| = \left\| \int S_q(r) \psi_{m,n}(r) r dr \right\| \quad (9)$$

Where $\psi_{m,n}(r)$ replaces $g_p(r)$ in (6), $m = 0, 1, 2, 3$, $n = 0, 1, \dots, 2^{m+1}$, $q = 0, 1, 2, 3$.

The above definition indicates that $\|F_{m,n,q}\|$ is actually a wavelet transform of $S_q(r)r$. For a fixed r , $S_q(r)$ represents the q th frequency feature of the image object $\text{Polar}(r, \theta)$ in the phase domain $[0, 2\pi]$. Furthermore, using different scale index

m and shift index n , the wavelet moment invariants $\|F_{m,n,q}\|$ can provide features of the object $Polar(r, \theta)$ at different scale levels.

From (6), we can see that $S_q(r) = \int Polar(r, \theta) e^{jq\theta} d\theta$, let $\Delta\theta = 2\pi / 256$. Then the discrete form of $S_q(r)$ is

$$S_q(\rho) = \frac{2\pi}{256} * \sum_{m=1}^{256} Polar(\rho, m) \cdot e^{-j2\pi(m-1)q/256} \quad (10)$$

Where $q = 0, 1, 2, 3$.

Based on (7) (8) (9) (10), we can easily get the discrete form of wavelet moment invariant features:

$$\|F_{m,n,q}\| = \left\| \frac{1}{32} \sum_{\rho=0}^1 S_q(\rho * 32) \psi_{m,n}(\rho) \rho \right\| \quad (11)$$

Where $\Delta\rho = 1/32$, $m = 0, 1, 2, 3$, $n = 0, 1, \dots, 2^{m+1}$, $q = 0, 1, 2, 3$, $S_q(\rho)$ is shown in (10).

C. Back-propagation on 3-layer MLP for classification

Artificial neural network based on MLP are feed forward nets with one or more layer of nodes between its input and output layers, and due to the nonlinearity activation function with each nodes, the MLP is capable of forming arbitrarily complex decision function in the pattern space. For a three layers MLP, the first layer is the input layer and represents the input values. The hidden layer is the second layer and is fully meshed to the previous input layer. The last layer is the output layer which receives values from the hidden layer.

The process of determining the weights is called training. The BP training algorithm is divided into two stages. In the first phase (forward process), it calculates the weights for each node layer by layer. In the second stage (back propagation process), it calculates error for each node of the hidden layer and fix previous layer weights with this error.

We take the sigmoid function as the activation function f , see (12).

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

In this paper, variables conventions are shown in Fig. 6. The input value of a neuron is the sum of the n neuron outputs of the previous layer, see (14). Take node j of the hidden layer as an example, node i stands for a node of the input layer, and node k stands for a node of the output layer. O_j is the output of node j , w_{ij} is the weight factor from node i to node j .

For the output layer, $\hat{y}_k = O_k$ is the actual output value, y_k is the expected output value, then the error of the current sample E and the error of all training samples E_{all} are

$$E = \frac{1}{2} \sum_k (y_k - \hat{y}_k)^2, \\ E_{all} = \sum_n E \quad (13)$$

Where n is the number of training samples. The network being convergent means E or E_{all} being confined to a limited value.

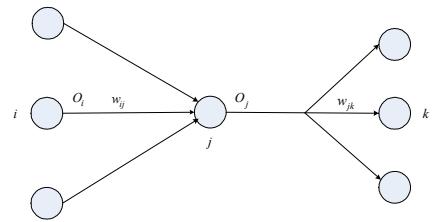


Fig. 6 Variables conventions of BP neural network.

The implementation steps of BP algorithm are as follows:

- 1) Initializing the weights to [-1, 1].
- 2) Repeat the next processes (in (14) ~ (18)) until the network is convergent or the training times reached the specified number (each sample is calculated respectively).

When inputting a new sample, calculate the outputs for each node layer by layer.

$$net_j = \sum_i w_{ij} O_i, O_j = 1/(1 + e^{-net_j}) \\ net_k = \sum_j w_{jk} O_j, O_k = 1/(1 + e^{-net_k}) \quad (14)$$

Calculate error δ_k for the output layer:

$$\delta_k = (y - O_k) O_k (1 - O_k) \quad (15)$$

Calculate error δ_j for the hidden layer:

$$\delta_j = O_j (1 - O_j) \sum_k w_{jk} \delta_k \quad (16)$$

Calculate the fixing values of $\Delta w_{jk}(t)$ and $\Delta w_{ij}(t)$ with error δ_k and δ_j .

$$\Delta w_{jk}(t) = \alpha \Delta w_{jk}(t-1) + \eta \delta_k O_j \\ \Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) + \eta \delta_j O_i \quad (17)$$

Where α is the inertial coefficient, and η is the training rate coefficient. η affects the convergence of the network, we can try to choose η from 0.01 to 1 to find the best value. α affects the training rate, which can be chosen from 0.9 to 1.

Fix previous layer weights with the fixing values $\Delta w_{jk}(t)$ and $\Delta w_{ij}(t)$.

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t) \\ w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (18)$$

The number of nodes in the hidden layer should be chosen according to experiments and the designer's experience. Too many hidden nodes will slow the training process and lower the classifying correct rate. The empirical formula below is provided for reference in choosing the best number of hidden nodes.

$$n_1 = \sqrt{n+m} + a \quad (19)$$

Where n_1 is the number of hidden nodes, n is the number of input nodes, m is the number of output nodes, a is a constant integer in the range [1, 10].

IV. EXPERIMENTS AND RESULTS

A. Translation normalization, scaling normalization and image size adjusting.

Fig. 7(a) is an original image of the tank object. Fig. 7(b), Fig. 7(c), Fig. 7(d) are the result images of translation normalization, scaling normalization and image size adjusting.

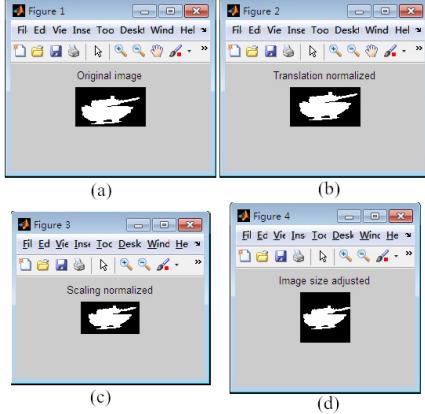


Fig. 7 Results of translation normalization, scaling normalization and image size adjusting.

B. Polar coordinate transformation.

In $\text{Polar}(r, \theta)$, the maximum of r (radius) is 32. θ is an integer, which divides 2π into θ parts. To find out the best value of θ , we conduct a series of experiments. Let θ be 256, 128 and 64 respectively, then $\Delta\theta$ is $2\pi/256$, $2\pi/128$ and $2\pi/64$. First take polar coordinate transformation of Fig. 7(d), and then take inverse coordinate transformation, we will get the “coordinate transformation—inverse coordinate transformation” result images shown in the first row of Fig. 8. Subtract Fig. 7(d) from the above result images and calculate the absolute values of the differences, the new result images are shown in the second row of Fig. 8. The figure shows that when θ equals 256, the error is negligible. When θ is smaller, the errors are obvious. So finally, we set $\theta=256$.

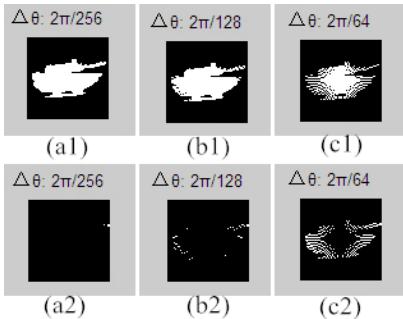


Fig. 8 Results of polar coordinate transformation.

C. Extracting wavelet moment invariants

As described in (11), the parameters of $\|F_{m,n,q}^{\text{wavelet}}\|$ are $m=0,1,2,3$, $n=0,1,\dots,2^{m+1}$ and $q=0,1,2,3$. In this paper, to screen out redundant and sensitive features and avoid oversampling, we select the features by another restriction of n : $n=0,1,\dots,2^m-1$. In this case, we can extract wavelet moment invariants with all together 44 dimensions.

In Fig. 9, we can see wavelet moment features of 3 similar tank objects. They are similar in shapes but different in details, which decides their wavelet moment feature curves also similar in shapes but different in details.

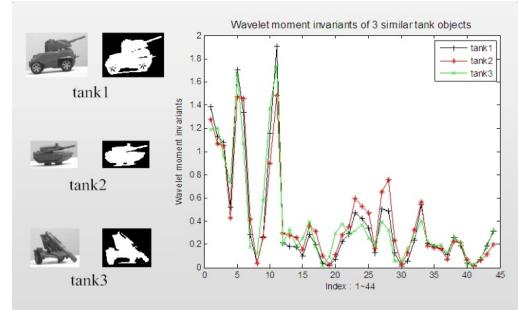


Fig. 9 Wavelet moment features of 3 similar tank objects.

D. Test the rotational, translational and scaling invariance of wavelet moment features.

Take turns to make rotational, translational, scaling transformation of an original image. Extract wavelet moment features of the four images, and then plot the curves of the features in Fig. 12. From the figure, we find the four curves almost completely overlap with each other, which proves the algorithm has rotational, translational and scaling invariance.

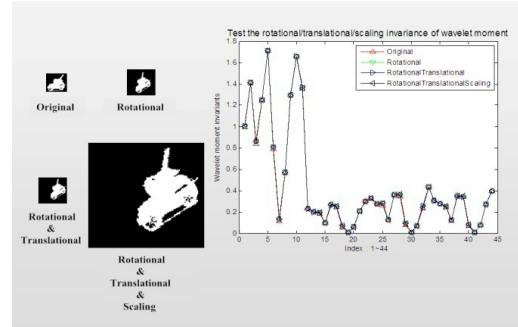


Fig. 10 Test the rotational/translational/scaling invariance of wavelet moment features.

E. Obtaining the training model.

We conduct off-line training and obtaining the classifiers of these two image sets. As discussed above, the number of input nodes equals the dimension of the wavelet moment features, i.e. 44, the number of output nodes equals the number of object kinds, and the number of hidden nodes is calculated according to (19). The first image set has 7 kinds of objects, so the number of hidden nodes is $n_1 = \sqrt{44+7} + a = 7.14+a \in [8, 17]$ (round the decimals). The second image set has 3 kinds of objects, so the number of hidden nodes is $n_2 = \sqrt{44+3} + a = 6.86+a \in [8, 17]$ (round the decimals). To observe more clearly, we expand the range [8,17] to [6,18]. When the number of hidden nodes varies from 6 to 18, the classifying correct rate curves of the two image sets are shown in Fig. 11. Picture (a) stands for the result of image set 1, while picture (b) stands for the result of image set 2.

From the figure, we can see that image set 1 reaches the highest correct rate 0.9236 when the number of hidden nodes

equals 15, and reaches the second – highest correct rate 0.9115 when the number of hidden nodes equals 8. The correct rate of image set 2 remains at 0.9837 when the number of hidden nodes varies from 6 to 18, only two images of class 1 are mistaken for class 3, and they are listed in Fig. 13 (a). It is clear to see that the two misclassified plane images are influenced by their shadows and seems like a man with his arms raised up, which leads to the misclassification.

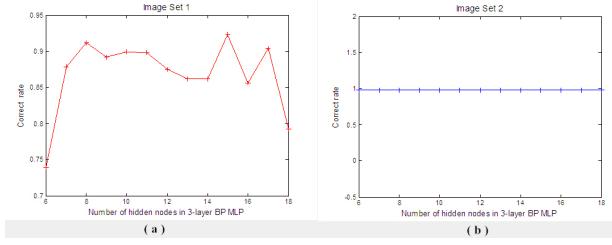


Fig. 11 Classifying correct rate curves of the two image sets.

For image set 1, when the number of hidden nodes equals 15, it is much slower in training speed than that the number of hidden nodes equals 8. Therefore, we choose 8 as the best number of hidden nodes, and the corresponding classifying result is shown in Fig. 12. For image set 2, the best number of hidden nodes is 6, and the corresponding classifying result is shown in Fig. 13(b). (i, j) is the probability of classifying class j into class i. The bold numbers in the diagonal are the correct classification rate of each kind. The overall correct rate is shown in the last row.

Image Set 1 (825 training images, another 825 testing images)								
(i, j): the probability of classifying class j into class i		j: true class						
		1	2	3	4	5	6	
i: result class	1	0.9055	0.0299	0.0104	0	0	0.0241	0.0114
	2	0.0597	0.8706	0.0104	0.0253	0.026	0	0
	3	0	0	0.9271	0.0127	0	0.012	0
	4	0	0.0697	0.0313	0.9494	0.013	0.012	0
	5	0.005	0.0149	0.0104	0	0.9221	0.012	0
	6	0	0.0149	0.0104	0.0127	0.013	0.8916	0.0114
	7	0.0299	0	0	0	0.026	0.0482	0.9773
Correct Rate								
0.9115								

Fig. 12 Classifying result of image set 1.

Image Set 2 (282 training images, 123 testing images)			
(i, j): the probability of classifying class j into class i		j: true class	
		1	2
i: result class	1	0.9512	0
	2	0	1
	3	0.0488	0
Correct Rate		0.9837	

(a) (b)

Fig. 13 Classifying result of image set 2.

F. Experiment and result for the embedded system.

Write the trained classifier into the DSP program, and then conduct real-time object detection and classification experiments on the image acquiring and processing board. Results are shown in Fig. 14. The letters C, M and P are marked in the image interface according to DSP recognition results. C stands for car objects, M represents man objects,

while P is on behalf of plane objects. We can see that the system can detect and recognize single object and multiple objects correctly.

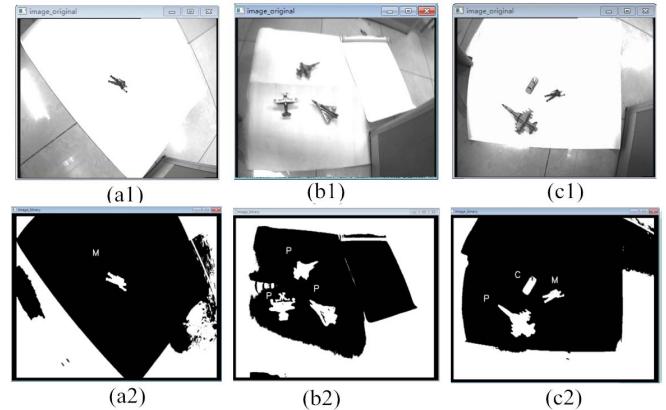


Fig. 14 Experiment and results for the embedded system

V. CONCLUSION

This paper proposed an algorithm of wavelet moment-based BP neural-network classifier on a low-power embedded image acquiring and processing board for real-time small immobile object detection and recognition. The proposed method is of good computation complexity and has scaling, translation and rotation invariance. Experiment results show that the method can detect and recognize single object and multiple objects of similar shape correctly and efficiently.

ACKNOWLEDGMENT

The work is supported by National Natural Science Foundation of China (No.61273360), and Key Project of S&T Plan of Beijing Municipal Commission of Education (No.KZ201210005001). Besides, the authors would like to thank the authors of references and the anonymous reviewers for improving the quality of this paper.

REFERENCES

- [1] Z. Wang, H. Song, et al, "A real-time small moving object detection system based on infrared image", Mechatronics and Automation (ICMA), 2014: 1149-1154.
- [2] R. DeFauw, S. Lakshmanan, and K.V. Prasad, "A system for small target detection, tracking, and classification", International Conference on IEEE/IEEJ/ISAI, pp. 639–644, October 1999.
- [3] T. Modegi, "Small object recognition techniques based on structured template matching for high-resolution satellite images", SICE Annual Conference, 2008: 2168-2173.
- [4] R. Manning, "Small object classification performance of high-resolution imaging sonars as a function of image resolution", OCEANS'02 MTS/IEEE, vol. 4, pp. 2156-2163, October 2002.
- [5] D. Shen, H.H.S. Ip, "Discriminative wavelet shape descriptors for recognition of 2-D patterns", Pattern recognition, vol. 32, no. 2, pp. 151–165, 1999.
- [6] M.K. Hu, "Visual pattern recognition by moment invariant" Information Theory, IRE Transactions, vol. 8, no. 2, pp. 179–187, February 1962.
- [7] M. Teague, "Image analysis via the general theory of moments", Opt Soc Amer, vol. 70, no. 8, pp. 920-930, 1980.
- [8] J. Skodzik, V. Altmann, et al, "A highly integrable FPGA-based runtime-configurable multilayer perceptron", Advanced Information Networking and Applications (AINA), pp. 429-436, March 2013.