

THE CLASSIFICATION MODEL OF SANDALS DHIIF'S PRODUCT WITH BACK PROPAGATION AND FOURIER DESCRIPTORS METHOD

Wikaria Gazali, Haryono Soeparno and Ngarap Im Manik

Department of Mathematics, School of Computer Science,
BINUS University Jl. K.H. Syahdan No.9, Jakarta 11480, Indonesia

Received 2012-12-28, Revised 2013-02-12; Accepted 2013-04-02

ABSTRACT

Classification is a vital process in every industry. While manual classification is inefficient in terms of cost and time, the computerized techniques offer better and faster solution on this matter. Fourier Descriptors is the mathematical formulas to save objects in functions of Fourier series. While back propagation is an Artificial Neural Network (ANN) method to recognize patterns and images through 'training'. Both of these methods will be used to produce a more efficient computerized techniques to the classification process itself.

Keywords: Classification, Back Propagation, Training, Pattern Recognition, Fourier Descriptors

1. INTRODUCTION

Product classification process is still done manually nowadays, especially in the small and medium industries. This mechanism is of course inefficient, because as production grows the company will have to hire more people. Therefore, a computerized technique will be very helpful in the attempt to make efficient the process itself. In this study the writer will explore the Pattern Recognition technology to recognize contours in order to classify products into categories. The writer has also developed an application to implement the technology into the real world (Tom, 1997).

This scope of this study will be confined to the recognition of two-dimensional-closed-curve-contour Bitmap (BMP) objects. The writer will use seven models of Dhiif's Sandals (the sandals will be analyzed individually), namely: Pandas, whales, dolphins, frogs, strawberries, watermelon and hearts and each model will have ten bitmap objects in different positions.

1.1. Fourier Descriptors of a Closed Curve

Consider a simple closed curve C arranged clockwise represented by $(x(I), y(I)) = Z(I)$ with the arc length l and $\Delta W_{ji}(n+1) = \eta \delta_j O_i + a \Delta W_{ji}(n)$. The direction of angle C is expressed by smooth Buck given $\delta_0 = \theta(0)$ as is the direction of the absolute angle at the initial point $Z(0)$. Cumulative angular function $\phi(l)$ is defined as a large angular corner of the network between the starting point and l like **Fig. 1**.

Therefore, the $\delta_0 = \theta(0)$ and $\phi(L) = -2\pi$, as all simple closed smooth curve with a clockwise orientation of the angular corner, has have a network of -2π . This can be seen in **Fig. 2**.

Region of origin of the hose $[0, L]$, of $\phi(l)$ has information and will be normalized into absolute size to the hose $[0, 2\pi]$ (a standard for periodic functions). Thus the normalized variance is defined by $\phi^*(t)$ such that Equation 1:

$$\phi^*(0) = \phi^*(2\pi) = 0 \quad (1)$$

Corresponding Author: Wikaria Gazali, Department of Mathematics, School of Computer Science,
BINUS University Jl. K.H. Syahdan No.9, Jakarta 11480, Indonesia

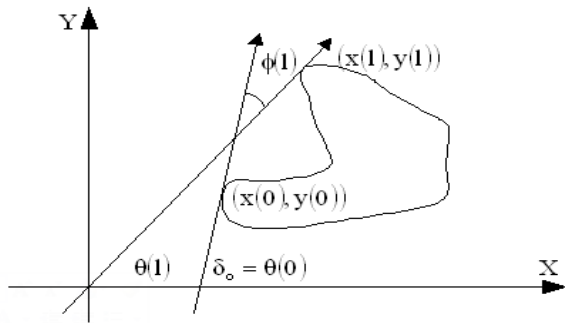


Fig. 1. Parametric representation of a closed plane curves with tangential angle $\theta(l)$ and the function of the bend is cumulative

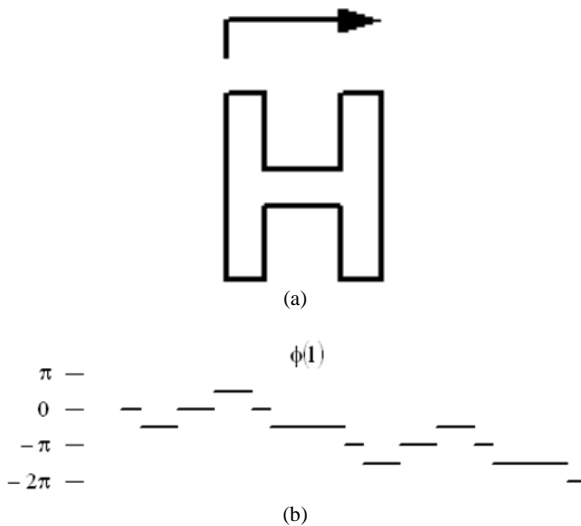


Fig. 2. (a) Closed plane curves with starting point of the character limit “H” (b) Cumulative angular bend function $\phi(l)$

The definition of normal is Equation 2:

$$\varphi^*(0) = \varphi\left(\frac{L}{2\pi}\right) + t \tag{2}$$

and φ^* is invariant under translation, rotation and change of circumference L. φ^* expanded into a Fourier series Equation 3:

$$\varphi^*(t) = \mu_0 + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt) \tag{3}$$

Or in polar becomes Equation 4:

$$\varphi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \tag{4}$$

where, (A_k, α_k) are the polar coordinates of (a_k, b_k) . A_k and α_k are the Fourier descriptors for curve C and each is called the k-th harmonic amplitude and phase angle to-k. If the polygon and curve C is given x_i, y_i for $i = 0, 1, \dots, N$ as the coordinates of N vertices and t_i as value parameters. Further $(x_N, y_N) = (x_0, y_0)$ and $t_N = t_0 + T$.

Fourier Transformation is given as Equation 5-7:

$$\begin{aligned} \begin{pmatrix} X_k \\ Y_k \end{pmatrix} &= \frac{T}{(2\pi k)^2} \sum_{i=0}^{N-1} \frac{1}{t_{i+1} - t_i} \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix} \\ &+ \frac{j}{2\pi k} \sum_{i=0}^{N-1} \frac{(x_{i+1} - x_i)}{(y_{i+1} - y_i)} \Phi_{k,i} \delta(t_{i+1} - t_i) \end{aligned} \tag{5}$$

Where:

$$\Phi_{k,i} = \exp\left\{-j 2\pi k \frac{t_i}{T}\right\} \tag{6}$$

And:

$$\delta(t_{i+1} - t_i) = \begin{cases} 1 & \text{jika } t_{i+1} = t_i \\ 0 & \text{jika } t_{i+1} \neq t_i \end{cases} \tag{7}$$

With the program design, it can be obtained by Fourier Descriptors parameters and a_n and b_n are obtained from the Fourier series (Arbter, 1989).

1.2. Back Propagation

Back Propagation is a multiple and interconnected layers neural network model with feed-forward architecture using a supervised learning method between layers.

Back propagation network model has been widely used in the development process of ANN for its good performance and accurate feedback. This model consists of two phases of training processes, the forward and backward propagation phase. When an input pattern is accepted by the first layer of the network, the input will be forwarded to the next layer. This process will occur on every layer until the final output is produced. The final output is then compared to the ideal output value resulting in error signal for each output. This signal will then be transmitted back from the output layer to each of the cells in the previous layer.

The network model itself consist of two mechanism or mode of operation which will work according to some specific rules, as mentioned below. This can be seen in **Fig. 3**.

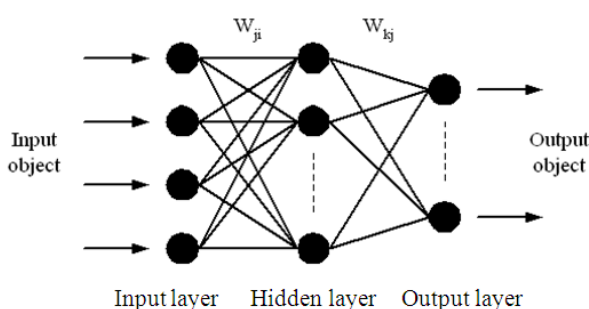


Fig. 3. The general scheme of non-linear neural network

1.3. Training Mode

In this mode, the network will be trained to be able to generate data in accordance with their intended target through one or more pairs of data input and output data. The longer workout or more pairs of data used for training the network the better the performance.

1.4. Production Mode/TryOut Mode

In this mode, the network is tested to determine whether it is able to recognize the data as expected.

Basically, there are three stages necessary to form a neural system, forward propagation, backward propagation and weight update

Input to the node network at layer j is Equation 8:

$$net_j = \sum_{i=0}^N W_{ji} O_i \tag{8}$$

The output of node j is Equation 9:

$$O_j = f(net_j) \tag{9}$$

where, f is the activation function. Function that rises monotonically and is not linear. For the training function must also be taken down and usually use the function signoida Equation 10:

$$O_j = \frac{1}{1 + e^{\frac{(net_j + \theta_j)}{\theta_0}}} \tag{10}$$

Also obtained Equation 11 and 12:

$$net_k = \sum_{j=0}^M W_{kj} O_j \tag{11}$$

And:

$$O_k = f(net_k) \tag{12}$$

In the learning phase of training such networks, given the input object $X_p = [x_1, x_2, \dots, x_n]$ as input, the network is asked to organize the set of weights in all the links that connect and also all of the threshold value in the node, such that the expected output is obtained at the output node t_{pk} . Once the adjustment is completed the network, given the couple x_p and others $\{t_{pk}\}$ and is asked to study the association's network. In fact the network was asked to find a set of weights and biases that will meet all of the input-output pairs are given. This process can be seen from the difficult learning process and even sometimes cannot be resolved easily. In general output $\{Q_{pk}\}$ will not be the same as the target or expected value $\{t_{pk}\}$. For each pattern, squared error is Equation 13:

$$E_p = \frac{1}{2} \sum_k (t_{pk} - O_{pk})^2 \tag{13}$$

By ignoring the index p for convenience, the equation can be written as Equation 14:

$$E = \frac{1}{2} \sum_k (t_k - O_k)^2 \tag{14}$$

Be expected Equation 15 and 16:

$$W_{ji}^{n+1} = W_{ji}^n + \Delta W_{ji} \tag{15}$$

And:

$$W_{kj}^{n+1} = W_{kj}^n + \Delta W_{kj} \tag{16}$$

Convergence is achieved towards the values for the weights and threshold are corrected by using Equation 17:

$$\Delta W_{kj} = -\eta \frac{\delta E}{\delta W_{kj}} \text{ dan } \Delta W_{ji} = -\eta \frac{\delta E}{\delta W_{ji}} \tag{17}$$

However, the average squared error E is expressed in the form of output O_k , each of which is non-linear output from node k (11) and net_k is output to node k and is defined as the sum of all weighted linear output at the previous layer (12).

And finally, is obtained Equation 18-20:

$$\Delta W_{kj} = \eta \delta_k O_j \text{ dan } \Delta W_{ji} = \eta \delta_j O_i \quad (18)$$

$$\delta_k = (t_k - O_k) O_k (1 - O_k) \quad (19)$$

$$\delta_j = O_j (1 - O_j) \sum_k \delta_k W_{kj} \quad (20)$$

There are several other issues to consider if you want to use the network. For example the question of how the value of h is selected. The value of h is proportional to learn quickly, but also can generate oscillations. Rumelhart, Hilton and Williams suggested that Equation (16) modified to contain the momentum of the order of the tribe.

So it can be written Equation 21:

$$\Delta W_{ji}(n+1) = \eta \delta_j O_i + \alpha \Delta W_{ji}(n) \quad (21)$$

where the quantity $(n+1)$ is used to show step $(n+1)$ and α is a constant proportional. The second term in Equation (19) was used to determine that a change in W_{ji} in step $(n+1)$ should be similar to changes that occur in the n -th step.

2. METATERIALS AND METHODS

The study makes use of Fourier Descriptors as a pre-process to create images that will be used as input for artificial neural networks. Process the image representation process with fourier descriptors consists of several stages, namely Picture Reading Stage, center of gravity searching stage, the stage of looking for Point M (farthest point), coordinate searching stage, the phase angle and the length of the contour of the object calculation and Fourier Descriptor parameters searching stage.

Results from the first five stages will enable us to calculate the parameters of the fourier descriptors of an object, which is as follow:

$$\mu_0 = -\pi - \frac{1}{L} \sum_{k=1}^m l_k \Delta \Phi_k$$

And Equation 22:

$$\begin{aligned} a_n &= \frac{-1}{n\pi} \sum_{k=1}^m \Delta \Phi_k \sin \frac{2n l_k \pi}{L} \\ b_n &= \frac{1}{n\pi} \sum_{k=1}^m \Delta \Phi_k \cos \frac{2n l_k \pi}{L} \end{aligned} \quad (22)$$

Where:

- m = Amount or number of points or vertices of an object contour
- l_k = Edge length of the k
- $\Delta \Phi_k$ = The angles at the vertex or point to the k
- L = Magnitude around the object (the object contour length)

The value of $m = 9$ is pre-determined, while the value of l_k and $\Delta \Phi_k$ (the length of the edge and corner) can be obtained from the procedure Hit_pjg_sdt. While the circumference of the object (value L) is obtained by summing the entire length of the edge on the contour of the object.

By knowing the magnitude of the values of variables are needed, the magnitude of the parameter of Fourier Descriptors can be searched or calculated. The parameters of the Fourier Descriptors of an object are the traits of the object. With these features, then an object can be recognized (Rao and Rao, 1996).

2.1. The Introduction of Image

ANN is used to classify the object image that has been represented as a Fourier series. Grouping process consists of two stages, namely training phase and testing phase. Stage of training is useful to determine the weights for each node in ANN layers, while the ANN model used is the Back Propagation neural network model with three layers topology, namely input layer, hidden layer and output layer.

In this study, 56 images are studied, thus we need 56 nodes in the input layer and seven types of sandals, thus we need 7 nodes in the output layer. The output will be compared with our expected result. If there is difference, then the difference will be back propagated to the previous layer to re-adjust the weights.

Basically, to form a neural system, it only needs three stages, namely the forward propagation, backward propagation and weight updates (Deutsch, 2011).

2.2. Screen Design

The program will be designed in a single main form, so that the user will be able to see the input status and the output in the same form without the need to flip through multiple displays. This can be seen in **Fig. 4**.

2.3. Implementation

The hardware and software specifications used to run this application are as follow:

- Processor: Pentium IV 1000 MHz

- Memory RAM: 256 Mbyte
- Video Graphic Adapter: NVIDIA RIVA TNT2 Model 64
- Monitor: AOC LM525A

As for software specifications, as follows:

- Operating system: Windows XP Service Pack 2
- Development environment: Microsoft .Net Framework 2.0
- Programming Language: Visual Studio C# 2005 Professional Edition

This module serves to display a brief description of applications such as titles, speakers and so on. The Enter key will bring the user to the second window, while pressing the Exit button, then the program will terminate. This can be seen in **Fig. 5**.

This module serves to display the image that will be tested on Picture Box Image and image recognition results in the Result. This can be seen in **Fig. 6**.

Furthermore, the user can download a Back Propagation Neural Network settings by selecting the Settings tab and fill in the required parameters. Henceforth, the application will automatically take a pre-existing conditions and the user can make changes to these provisions.

In this case the author uses three layers, namely:

- Input layer = 56
- Hidden layer = 10
- Output layer = 7

Network that was built is stored in file 'train.net'.

Testing process or can also be referred to as the process of determining the image recognition. This stage aims to process the output of the ANN or recognize the patterns generated by the test module to generate the image pattern, with reference to the learning outcomes conducted during the training process. To begin testing, the user can press the button "Browse" active immediately after the parameter of Fourier descriptors obtained. This will bring up the Open Dialog Box and the user prompted to insert the image will be tested as shown in **Fig. 6**.

Once selected button "Open" then the image files will appear on Picture Box Image. Furthermore, the user may choose to provide a training network in accordance with the parameters on the Settings tab select the button "Train Network" or on a network that has been stored before selecting the button "Load Network". After the training process or the loading network is successful, then the button "Recognize" will be active and ready to process the image to be tested. Image recognition results will be displayed on the Result and the percentage similarity. This can be seen in **Fig. 7**.

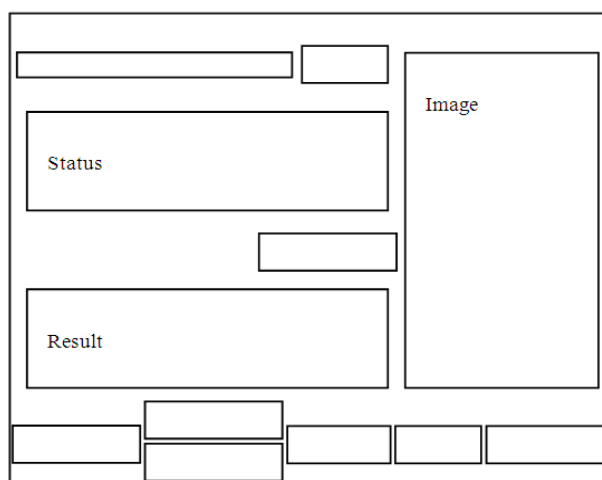


Fig. 4. Designing the main form

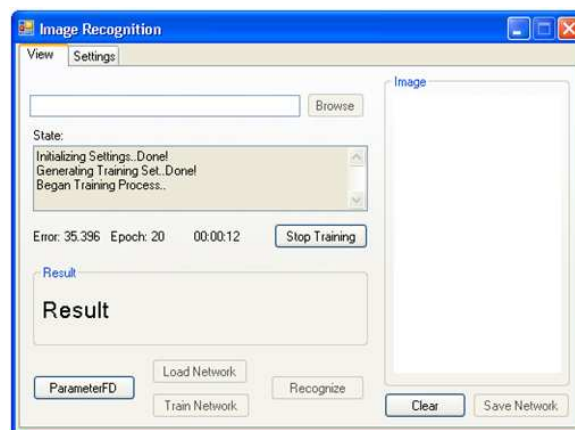


Fig. 5. Display form view

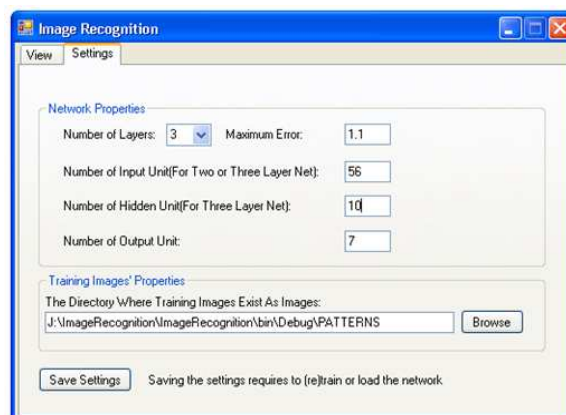


Fig. 6. Display form setting



Fig. 7. Results of test image



Fig. 8. The Pattern of Sandals

3. RESULTS AND DISCUSSION

To test the application program, in this study used 70 images of objects consisting of 7 Dhiif sandals contour model's looks on (right or left, not a pair). Each model sandals sample of 10 images taken in different positions, in which eight images are used as a training phase (learning sample) and the second image is used as the testing phase (sample detection). This can be seen in Fig. 8.

After the training process image data on 56 successful, then the 56 images were tested again on Image Recognition software applications via the mechanism of the system work as described in the previous section to find out whether the program can serve the purpose and at what level of accuracy. This can be seen in Table 1.

Table 1. Table of results the introduction of image that has been trained

Image	Known as
01heart.bmp	heart
02heart.bmp	heart
03 heart.bmp	heart
04 heart.bmp	heart
05 heart.bmp	heart
06 heart.bmp	heart
07 heart.bmp	heart
08 heart.bmp	heart
01frog.bmp	frog
02frog.bmp	frog
03frog.bmp	frog
04frog.bmp	frog
05frog.bmp	frog
06frog.bmp	frog
07frog.bmp	strawberries
08frog.bmp	frog
01dolphin.bmp	dolphin
02dolphin.bmp	dolphin
03dolphin.bmp	dolphin
04dolphin.bmp	dolphin
05dolphin.bmp	dolphin
06dolphin.bmp	dolphin
07dolphin.bmp	dolphin
08dolphin.bmp	dolphin
01panda.bmp	panda
02panda.bmp	panda
03panda.bmp	panda
04panda.bmp	panda
05panda.bmp	panda
06panda.bmp	panda
07panda.bmp	panda
08panda.bmp	panda
01whale.bmp	whale
02whale.bmp	whale
03whale.bmp	whale
04whale.bmp	whale
05whale.bmp	whale
06whale.bmp	strawberries
07whale.bmp	whale
08whale.bmp	whale
01watermelon.bmp	watermelon
02watermelon.bmp	watermelon
03watermelon.bmp	watermelon
04watermelon.bmp	watermelon
05watermelon.bmp	watermelon
06watermelon.bmp	watermelon
07watermelon.bmp	watermelon
08watermelon.bmp	watermelon
01strawberries.bmp	strawberries
02strawberries.bmp	strawberries
03strawberries.bmp	strawberries
04strawberries.bmp	strawberries
05strawberries.bmp	strawberries
06strawberries.bmp	strawberries
07strawberries.bmp	strawberries
08strawberries.bmp	strawberries

Table 2. Results of position goods

Image	Known as
01heart_test.bmp	heart
02heart_test.bmp	heart
03frog_test.bmp	strawberries
04frog_test.bmp	frog
05dolphin_test.bmp	strawberries
06dolphin_test.bmp	dolphin
07panda_test.bmp	panda
08panda_test.bmp	whale
09watermelon_test.bmp	watermelon
10watermelon_test.bmp	watermelon
11strawberries_test.bmp	strawberries
12strawberries_test.bmp	strawberries
13whale_test.bmp	whale
14whale_test.bmp	strawberries

Apart from conducting tests on images that have been trained, testing is also done on 14 new images that have not been trained through the system the same mechanism as described in earlier to find out how effective the application of Image Recognition software is in recognizing the image, especially for new image who had never trained. This can be seen in **Table 2**.

4. CONCLUSION

Based on the results of the analysis we can conclude that: From the show-case application we may conclude the we gain a satisfactory results with 91.43% level of accuracy.

5. REFERENCES

- Arbter, K., 1989. Affine-invariant fourier descriptors, in from pixel to features. Pixels Features.
- Deutsch, J.A., 2011. A theory of shape recognition. Br. J. Psych, 46: 30-37. DOI: 10.1111/j.2044-8295.1955.tb00521.x
- Rao, V.B. and H.V. Rao, 1996. C++ Neural Network and Fuzzy Logic. 2st Edn., BPB Publications, New Delhi, ISBN-10: 8170296943, pp: 551.
- Tom, M.M., 1997. Machine Learning. 1st Edn., McGraw-Hill, Inc. Singapore.