# Method for Estimation of Crowd Density Using Neural Network with PSO Optimization Based on Gray Level Co-Occurrence Matrix

Xie Lili and Wang Peng

***Abstract*—In the paper, a method was proposed to estimate population density based on the gray level co-occurrence matrix. Builded the relation between contrast and population density, and designed a three layers neural network. This paper has used particle swarm algorithm(PSO) to train and optimize the neural network's connection weights and thresholds, which can overcome the BP algorithm's limitations such as slow convergence and easy to fall into local minima. The experimental results indicated that this method is fast and can effectively calculate the crowd density. This paper first expounded the research method of crowd population density; then introduced the method of gray level co-occurrence matrix and established the three layer neural network model; the model was optimized by back-propagation (BP) and particle swarm algorithm (PSO); finally, the paper compared The optimization results of BP algorithm with PSO algorithm, And drew the conclusion.**

***Index Terms*—Crowd density, neural network, particle swarm algorithm, gray level co-occurrence matrix.**

## I. Introduction

With the development of human society and progress of science, statistics for population density of public places, such as airports, stations, shopping alls, school buildings has become an important issue for city safety management field. Relevant institutions can make appropriate security solution according to the statistical results. Now, the application of computer vision and intelligent algorithm for statistics and estimation of the population density has become a research focus.

The present study method on population density can be divided into two categories:

### A. Sensor Identification

This method is mainly used in access control system. The pressure sensors can be put on the bus door downward, which can count the number when the passengers get on the bus. As everyone knows, radio frequency identification identification device (RFID) is widely used in teaching building, library etc. The advantages of sensor method are accurate, fast but it has higher cost and is not suitable for crowded places.

### B. Image Analysis Method

This method use intelligent algorithm to calculate population density based on the analysis of real-time images collected by video surveillance equipment. Image analysis needs less hardware facilities, lower cost, and easy integration. This method has some errors. But with the application of various intelligent algorithm progress of science, errors will be smaller and smaller. Image analysis method can be divided into two kinds, one is human body characteristic analysis and the other is estimation method, according to the analysis objects [1], [2].

Analysis of human body characteristic is to identify and statistical characteristics of the human body image, such as the recognition of facial information [3], [4].The method is suitable for small scene and fewer occasions. Advantages of the method are accurate; Disadvantage are controlling complexly and slowly. For the big and dense scene, there are another method to estimate the crowd density.

Crowd density estimation method is carries on the analysis to the training samples, determining the correspondence between features and population density in the image [5], [6]. This method is the focus of current research, this paper is to estimate the population density based on the analysis of image features.

The main work of this paper is summarized as follows:

Calculate each gray level co-occurrence matrix (GLCM) on four directions of image sample. Calculate the contrast for each GLCM. Each image get four contrast.

Design a three layers neural network, which use the four contrast values as the input. The output of the neural network is the density estimation we want.

Build Particle Swarm Optimization (PSO) model. Samples were used to train the PSO model. A particle vector is made by connection weights and thresholds of the neural network, which is trained by samples.

Calculate the population density. With the test sample is input to the neural network, which is optimized by PSO algorithms in step three, we can get the population density.

## II. GLCM and Characteristics of Image

In this paper, we use image's texture information to estimate the crowd density [7]. Studies show that the crowd image with high density has fine texture and the low density image has coarse texture [8]. For the texture is composed of gray distribution which is repeatedly formed in image space, there will be some relationships-the image gray spatial correlation properties between two pixels that are within a certain distance. GLCM is a common method to describe the

Manuscript received September 3, 2013; revised October 30, 2013. This project is supported by Tianjin Science Education "The Twelfth Five-Year Guideline" (No.VEYP6011).

Xie Lili is with Tian Jin Vocational Institute, Tianjin 300410 China (e-mail: shelly0313@126.com).

Wang Peng is with Hebei University of Technology, Tianjin, 300401, China (e-mail: Wangpeng1027@126.com).

DOI: 10.7763/IJMLC.2013.V3.373

texture through the gray spatial correlation properties [9].

In order to calculate the GLCM, we can take a couple points A($x$, $y$) and B($x$+$a$, $y$+$b$) whose grey level is ($g$1, $g$2) . Make the point A ($x$, $y$) move across the whole image, we will get all kinds of ($G$1, $G$2) values. If a grey image has $N$ levels, there will be N2 combinations of ($g$1, $g$2). For the whole image, statistics for occurrence frequency of each ($g$1, $g$2) value and then arranged in a matrix (N×N). This matrix is called GLCM.

Commonly used measurements for GLCM include contrast, homogeneity, angular second moment etc, in which contrast is the most impotent one.

$$f_c^{(q,d)} = \sum_{i=1}^{N} \sum_{j=1}^{N} (i - j)^2 P(i, j)^{(q,d)} \tag{1}$$

In formula, ($\theta$, $d$) is direction and distance between two pixels. $N$ is the dimension of the GLCM. $P$ ($I$, $J$) is the value of point ($i$, $j$) in matrix.

Research shows that， in different directions, 0 °, 45 °, 90 ° and 135 ° (counter clock wise) have the most obvious contrasts. Four directions are showed in Fig. 1.

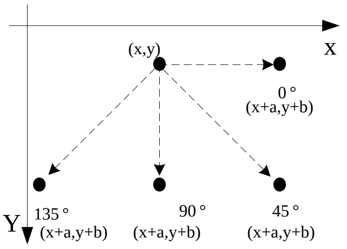


Fig. 1. four direction of GLCM.

To calculate the GLCM's contrast in different directions, we need process each image sample as follows:

### A. Gray-Scale Image Processing

Each gray level value of pixel is calculate according to the following formula:

$$g(i, j) = \left(\mathrm{Re}\,d(i, j) + Green(i, j) + Blue(i, j)\right)/3 \tag{2}$$

### B. Get Four Gray Level Co-Occurrence Matrixes of Four Directions

Out = cooccurrence (input, dir, dist, symmetric);

Input, is the gray matrix of image calculated by step 1;Dir is the direction which value can be set 0, 1, 2, 3 corresponding to 0°, 45°, 90°and 135°;Dist is the distance parameter, this paper set 1;

### C. Calculate Each GLCM's Contrast and Normalized

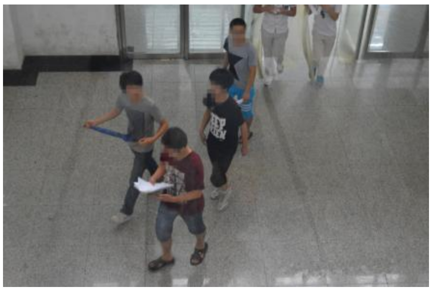


Fig. 2. The original color image.

Finally, each image will obtain four contrast values.

Fig. 2 is a test sample (Taken from a teaching building hall in Tianjin Vocational Institute). Fig. 3 is after the gray processing. Four directions GLCM is shown in Fig. 4.

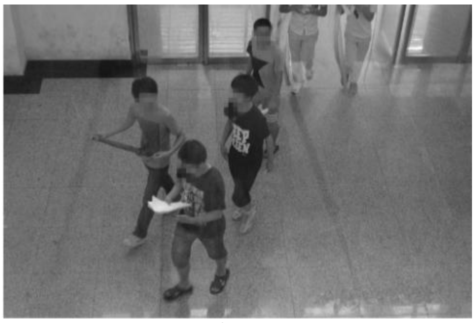
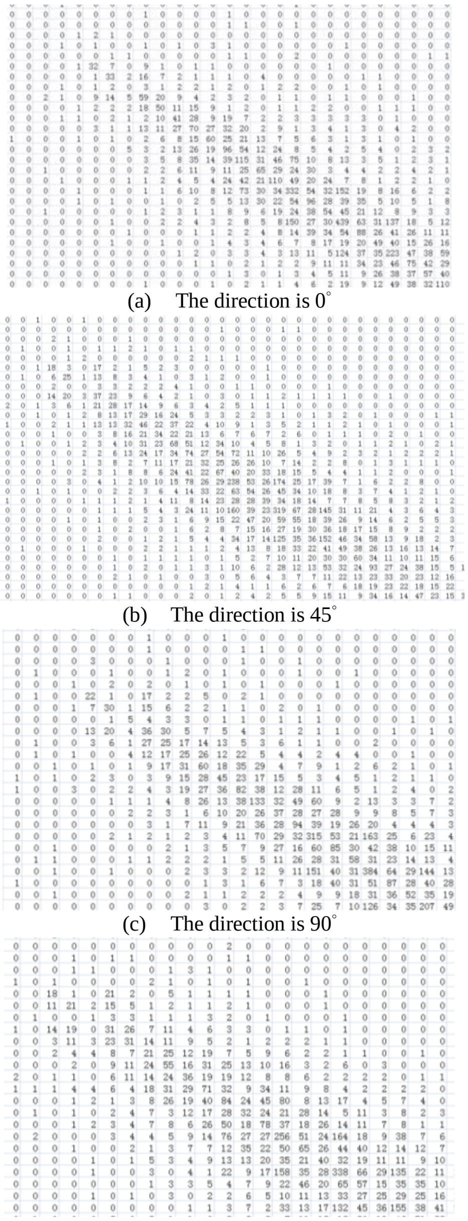


Fig. 3. The image after the gray processing.



(a) The direction is 0°

(b) The direction is 45°

(c) The direction is 90°

(d) The direction is 135°

Fig. 4. Part of the GLCM.

There are fewer people in the image, corresponding to the rough texture. Gray level co-occurrence matrix's values are mainly distributed in the vicinity of the diagonal which make a relative small contrast. Fig. 3 Contrast of four direction values as shown in Table I.

TABLE I: FOUR CONTRAST OF FIG. 3

| Direction | 0 ° | 45 ° | 90 ° | 135 ° |
|---|---|---|---|---|
| Contrast | 85.336 | 126.8664 | 65.7940 | 118.9844 |

## III. THE MODEL FOR CROWD DENSITY ESTIMATION

### A. *Artificial Neural Network Model*

This paper used the neural network to estimate crowd density. We designed a network model including three layers, input layer, hidden layer and output layer, showed in Fig. 5.

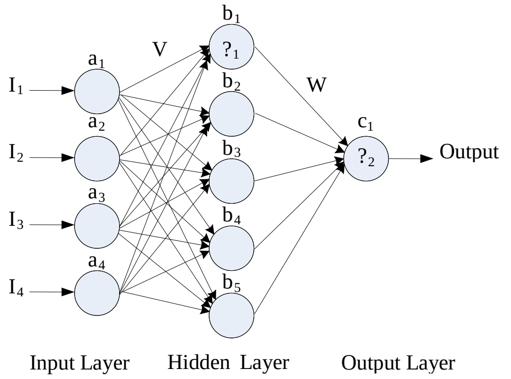


Fig. 5. neural network model.

The input layer had 4 nodes corresponding to four contrast of four directions. The hidden layer had 5 nodes. Output of the network will be the crowd density.

The formulas for hidden layer node and output node are:

$$H_j = f\left|\sum_{i=1}^{m} v_{ji} I_i + \theta_j\right| \tag{3}$$

$$O_k = f\left|\sum_{j=1}^{n} w_{kj} H_j + \theta_k\right| \tag{4}$$

In formula 3, $H_j$ is the output value of hidden lay node *j*; In formula 4, $O_k$ is the output value of output node *j*; *M*(=5) and *n*(=1) are the node numbers of input layer and output layer ; The $v_{ji}$ is connection weight of input node *i* and hidden layer node *j*; The $w_{kj}$ is connection weight of output node *k* and hidden layer node *j*; $I_i$ is input value of node *i*; The $\theta_j$ is threshold of hidden layer node *j*; The $\theta_k$ is threshold of output layer node *k*; The *f* is S type activation function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{5}$$

### B. *Particle Swarm Optimization Algorithm (PSO)*

PSO algorithm is a heuristic method, which simulate the foraging behavior of birds [10]. In the PSO algorithm, the solution of optimization is the particle position in the search space. All particles calculate the adaptation through an optimization objective function. Each particle has a velocity parameter to determine the direction and distance of their search. Each particle is updated by tracking the local extremum extremism and the globe extremum.

Assuming in a D dimensional search space, there are M particles consist of a particle group, in which the position of the I particle, $X_i = (x_{i1}, x_{i2}, ..., x_{iD}), (i = 1,2,...,M)$, is a potential solution to the optimization problem.

The velocity value of particle *i* on all dimension are recorded as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$ .The best location for particle *i* in history is $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$ .The best historical location for the whole swarm is $P_g = (p_{g1}, p_{g2}, ..., p_{gD})$ .When find out two best locations, each particle will update the speed and position according to the following formula:

$$v_{id}^{(t+1)} = u \cdot v_{id}^{t} + c_1 r_1 (p_{id} - x_{id}^{t}) + c_2 r_2 (p_{gd} - x_{id}^{t}) \tag{6}$$

$$x_{id}^{(t+1)} = x_{id}^{t} + v_{id}^{t+1} \tag{7}$$

In the formula, u is the inertia weight, usually set to for 0.4~1.2. The $c_1$ and $c_2$ are called acceleration constant or learning factor, usually $c_1 = c_2 = 2$ .The $r_1$ and $r_2$ are two random numbers changes in [0, 1].

In this paper, we calculated the connection weights and threshold of neural network using PSO algorithm.

## IV. EXPERIMENT RESULTS ANALYZE

This paper used a total of 200 pictures, including 100 pieces to train neural network using the PSO algorithm, and 100 pieces for the test. The hidden layer of neural network included 4×5=20 connection weights. The output layer contained 5 weights. The hidden layer and the output layer had a total of 6 thresholds. Corresponding, the particle dimension is 31, *X*=(*v*1 …*v*20 *w*1 …*w*5 *θ*1 … *θ*6). The number of particles was 100.

Fitness function:

$$g(O, A) = |O - A| \tag{8}$$

In the formula, O was the actual output value and A was the expected output value of the neural network.

The goal of PSO algorithm is to find such a particle coordinates (neural network weights and thresholds), which make the fitness function (Equation (8)) get the minimum value.

The steps for PSO algorithm are:

Initialize the velocity ($V_i$) and coordinate ($X_i$) for each particle swarm. Set $P_i = X_i$. According to the formula (8), calculate the fitness of each particle and set $P_g$ to the minimum fitness degree particle's coordinate.

In each cycle, calculate fitness for each particle and update the coordinate $P_i$ if the new fitness is better than the old fitness. Find the smallest fitness particle to update the $P_g = (p_{g1}, p_{g2}, ..., p_{gD})$ .

Each particle updates the speed and position according to the following formula (6) and (7).

The process cycle until meet the goal of adaptive error or meet the maximum training counts.

The Grading standard of population density is given in Table II.

TABLE II: THE GRADING STANDARD OF POPULATION DENSITY

| People/2m2 | 0~2 | 3~4 | 5~6 | 7 and more |
|---|---|---|---|---|
| Density | D⩽0.25 | 0.25<D⩽0.5 | 0.5<D⩽0.75 | D>0.75 |
| Remark | sparse | Low | middle | high |
| Standard | 1 | 2 | 3 | 4 |

For each training sample, the level of congestion l selects 25 images, so there are totally 100 images as a sample set. Training iteration number is set ed 100 and the error is set ed 10-2, the training process is shown in Fig. 6 and Fig. 7.

As shown in Fig. 6 and Fig. 7, the weight and the threshold value of PSO algorithm for neural network training is feasible, and the training numbers of PSO algorithm is less than BP neural network.

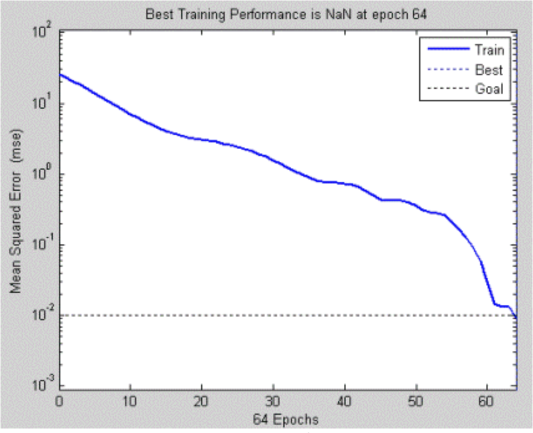


Fig. 6. The training process of BP neural network.

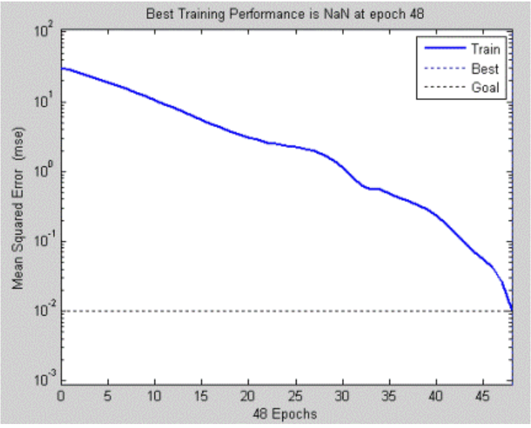


Fig. 7. The training process of PSO algorithm.

TABLE III: COMPARISON OF TESTING RESULTS

| Sample level | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| congestion level | sparse | low | middle | high |
| sample size | 25 | 25 | 25 | 25 |
| The number of errors | 5 | 3 | 1 | 2 |
| error | 20% | 12% | 4% | 8% |
| overall energy | 11% | | | |

The testing degree of congestion sample is calculated using the weights and thresholds of neural network by PSO algorithm. The comparison of the test sample and actual data is shown in Table III.

From the test results, the calculation error of this method is slightly wide in the sparse and low density area, and minor in denser and high density area. The average error is small.

## V. CONCLUSIONS

This paper is based on Gray level co-occurrence matrix. The contrast was used as the indicator of calculating population density. The neural networks of three layer models were established and optimized the weight and the threshold value using PSO algorithm. The experimental results show that the PSO algorithm is not only simple, small amount of calculation, but also can accelerate the training speed, to avoid falling into local minimum. For the testing samples, the model of crowd density is based on PSO algorithm can estimate well and be less average error. Future work was focused on the high density range and a more precise calculation.

## REFERENCES

[1] Q. Tang, Z. Y. Wang, H. P. Yan *et al.*, "Crowd density estimation of wide scene based on fuzzy neural network," *The Research and Application of Computer*, vol. 27, no. 3, pp. 989-991, 2010.

[2] S. Guo, H. P. Yan, and W. Liu, "Crowd density estimation based on grey level dependence matrix," *Computer Engineering and Applications*, vol. 44, no. 28, pp. 169-171, 2008.

[3] W. H. Ma, L. Huang, and C. P. Liu, "Advanced local binary pattern descriptors for crowd estimation," in *Proc. IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008.

[4] X. H. Qin, X. F. Wang, X. Zhou *et al.*, "Counting people in various crowed density scenes using support vector regression," *Journal of Image and Graphics*, vol. 44, no. 28, pp. 169-171, 2008.

[5] G. Q. Yang and R. Y. Cui, "Crowd density estimation based on probabilistic neural network," *Journal of Yanbian University (Natural Science)*, vol. 36, no. 3, pp. 250-253, 2010.

[6] D. Conte, P. Foggia, and G. Percannella, "A method for counting peoplein crowded scenes," in *Proc. the Seventh IEEE International Conference on Advanced Video and Signal based Surveil-lance*, 2011.

[7] A. N. Marana, S. A. Velastin, and L. F. Costa *et al.*, "Automatic estimation of crowd density using texture," *Safety Science*, vol. 28, no. 3, pp. 165-175, 1998.

[8] L. Chen, J. Tao, Y. P. Tan *et al.*, "People counting using iterative mean-shift fitting with symmetry measure," in *Proc. the 6th International Conference on Information and Communication Security*, 2007, pp. 890-895.

[9] W. Q. Zhang, "Crowd density estimation based on grey co-occurrence matrix and fractal," *Electronic Test*, vol. 4, pp. 36-39, 2012.

[10] G. B. Secrestbr Lamont, "Visualizing particle swarm optimization," in *Proc. the 2003 IEEE Swarm Intelligence Symposium 'SIS'03*, Indianapolis (IN，USA), pp. 2003.

[11] B. R. Secrest and G. B. Lamont, "Visualizing particle swarm optimization-Gaussian particle swarm optimization," in *Proc. the Swarm Intelligence Symposium*, 2003, pp. 198-204.

**Xie Lili** was born on 1978. She received a bachelor's degree in Ordnance Engineering College in 1997 and received a master's degree from Hebei University of Technology in 2001. Now she is a lecturer in Tianjin Vocational Institute. Since she served as a lecturer, she published 7 papers successively by the first author, one of the papers included in EI. She presided over the provincial and ministerial level projects 2 and participate in the provincial project 2.