Expert Systems with Applications 36 (2009) 5212–5220

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Image backlight compensation using neuro-fuzzy networks with immune particle swarm optimization

Cheng-Jian Lin [a,*], Yong-Cheng Liu [b]

[a] Department of Computer Science and Engineering, National Chin-Yi University of Technology, Taichung County 411, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung County 413, Taiwan, ROC

ARTICLE INFO

Keywords:
Backlight compensation
Neuro-fuzzy networks
Immune algorithm
Particle swarm optimization

ABSTRACT

In this study, we proposed a new technique to compensate the backlight images. Two processing stages, called the backlight level detection and the backlight image compensation, are proposed. In the backlight level detection stage, we first transferred the color space to gray space by feature weighting, then obtain two backlight factors. We apply these two backlight factors to the proposed functional-link-based neuro-fuzzy network (FNFN) with immune particle swarm optimization (IPSO) for detecting compensation degree. In the backlight image compensation stage, we also proposed the adaptive cubic curve method to compensate and enhance the brightness of backlight images according to the compensation degree of each image. The backlight degree is indicated by histograms of the luminance distribution in the backlight level detection stage. The experiment results showed that the backlight images can be compensated effectively.

1. Introduction

Digital cameras (DC) are getting more universal in our daily. They even become one of the most essential functions for mobile phones. Recently, the techniques of the digital camera have many appealing features, such as auto focus, auto exposure, etc. However, users still have chances of getting backlight images, the backlight problem needs to be improved. In this study, we mainly focus on addressing an effective compensation method for the backlight problem.

Recently, some researches about backlight image compensation have been reported, but the numbers are very scanty (Chin & Lin, 2005; Lin & Huang, 2003; Shimizu, Kondo, Kohashi, Tsumta, & Komuro, 1992). Shimizu et al. (1992) proposed an algorithm to compensate exposure in the case of backlighting or excessive front-lighting, regardless of the position of objects. To achieve this compensation, not only a new criterion parameter called HIST is introduced in the system but also fuzzy logic is applied to determine the amount of compensation. Lin and Huang(2003) proposed a two-stage compensation technique for improving the appearance of the pictures. They utilize the fuzzy c-means learning mechanism and the fuzzy logic rule inference to compensate the back-light images. To overcome the disadvantages of conventional backlight image processing methods, such as over-saturation and diminished contrast, Chin and Lin (2005) proposed a backlight image detection and compensation algorithm with fuzzy logic and adaptive compensation curve. In Lin and Huang (2003) and Chin and Lin (2005), they are also use fuzzy logic and two backlight factors to determine the compensation value of backlight images, but they employed difference ways to extract the two backlight factors from backlight images.

In this article, there are two major stages, which are the backlight level detecting and backlight image compensation. First, in the backlight prediction step, we will transfer images to gray value, then use pixel for the clustering which will separate the backlight object and the background. We can gather two backlight factors from the information which is in result of the clustering and the gray histogram. We apply these two backlight factors to the proposed functional-link-based neuro-fuzzy network (FNFN) (Chen, Lin, & Lin, 2007), for image compensation. We also proposed a new learning algorithm, called the immune particle swarm optimization (IPSO) (Lee, Lin, & Chuang, 2006), to train the FNFN model. Secondly, according to the deductive compensation value of the FNFN in the backlight compensation stage, we can gather a compensation curve from the backlight image. After the backlight image is transferred through the curve function, it will be acquired a compensation image. Then following by applying this image with this technique and obtain a backlight compensation, we will not have to worry about the backlight problem which is caused by the main accessories between the light and the lens, when the photos are taken. This is because all backlight images can be compensated. In our simulations, we have actually applied above-mentioned techniques to solve backlight image problems.

* Corresponding author.
E-mail address: cjlin@cyut.edu.tw (C.-J. Lin).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.06.109

The proposed method performs the accurate detection of the backlight degree and strong compensation effect in backlight images.

The rest of this paper is organized as follows. Section 2 describes the proposed compensation method. Next, Section 3 presents the simulation results of several backlight images. Conclusions are finally drawn in Section 4.

2. The proposed compensation method

In this section, we will introduce the proposed approach for backlight image compensation. The flow chart of the proposed algorithm is shown in Fig. 1.

2.1. Image color space transformation

In order to satisfy the follow-up two unit: the image histogram and the region clustering, we change images color space from the RGB color model to the YIQ model, and we adopt to operate on the luminance Y component only. This design choice takes advantage of the fact that the human eyes are less sensitive to quantize errors affecting the chrominance components of the image. Eq. (1) shows the color space transformation from RGB to YIQ.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix}. \tag{1}$$

2.2. Determine two factors of the backlight image

2.2.1. Backlight factor B_fcm

In a backlight image, the contrast between background and backlight objects is usually very great. Through observation of the backlight image histogram, we can find a phenomenon, the distribution of the luminance between background and backlight object that presents the obvious distance. Fig. 2 shows a backlight image and its histogram.

Therefore, in order to measure the gradient between background and backlight object in HIST histogram, we utilize sliding window (SW) to deal with the *B*_hist factor. HIST is defined as the ratio between the number of pixels whose brightness is higher than a threshold value and the total number of pixels in the whole image. First, we apply the SW to calculate the max SW, when the accumulation of the HIST is smaller than 0.2. Fig. 3 show the HIST histogram and the max SW:

We can find a phenomenon through observing, the backlight degree will raise with the $SW_{\max}$. So, we define the *B*_hist as following equation:

$$B\_\text{hist} = T\_\text{hist}\left(\frac{SW_{\max}}{255}\right), \tag{2}$$

where $T_{\text{hist}}(.)$ was a transfer function which transformed *B*_hist into a fuzzy degree. The transfer function is show in the list:

$$T_{\text{hist}}(x) = \begin{cases} (x-0.3)/(0.6-0.3), & \text{if } 0.6 > x > 0.3, \\ 1, & \text{if } x >= 0.6, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

2.2.2. Backlight factor B_fcm

Another backlight factor, we also use the cluster result extract by fuzzy c-means (FCM) (Pal & Bezdek, 1995) to determine it. Separately, the background and backlight object represent the bright and dark area separately in backlight images. Therefore, for the sake of segmenting background and backlight object, the cluster number is set to 2. After FCM algorithm, we will obtain two clustering centroid C1and C2. According to the image histogram information, the luminance can‘t be accumulated in histogram area between the background and the backlight object, In other word, the smaller accumulating amount of luminance between the background and backlight object is, then the higher the backlight degree is. Based on the characteristic of the above, we can define another backlight factor, *B*_fcm for determining the backlight degree of an image:

$$\text{B}_{\text{fcm}} = T_{\text{fcm}}\left(\frac{\sum_{i=C_1}^{C_2} p(r_i)}{(C_2 - C_1) * \frac{p(C_1)+p(C_2)}{2}}\right) \tag{4}$$

where $p(r_i) = \frac{n_i}{n}$ is the probability of the *i*th gray level, when n was the total number of pixels in the image and n_i was the number of times the level appeared in the image, $T_{\text{fcm}}(.)$ was a transfer function which transformed *B*_fcm into a fuzzy degree. The transfer function is show in the list:

$$T_{\text{fcm}}(x) = \begin{cases} (0.8-x)/(0.8-0.25), & \text{if x gt; } 0.8 \\ 0 & \text{if } x > 0.8, \\ 1, & \text{otherwise}. \end{cases} \tag{5}$$

The factor, *B*_fcm, expresses the accumulation of the luminance between two cluster centroid *C*1and *C*2, and the backlight degree is growing with *B*_fcm.

2.3. A new functional-link-based neuro-fuzzy network

Recently, neuro-fuzzy networks have been demonstrated in lots of research (Halgamuge, 1998; Juang & Lin, 1998; Lin & Lin, 1997; Lin & Xu, 2006; Patra & Pal, 1995; Patra, Pal, Chatterji, & Panda, 1999; Takagi & Sugeno, 1985). Neuro-fuzzy network owns the advantage of fuzzy system and neural network simultaneously: one is the inference characteristic of the fuzzy system; the other one is the system according to the learning ability of the neural network which can do the adjustment of the fuzzy rule. Therefore, the neuron-fuzzy network becomes a popular research target progressively, and applies on various problems, such as control, prediction, classification and pattern recognition. Two typical types of neuro-fuzzy networks are Mamdani-type (Lin & Lin, 1997; Halgamuge, 1998) and TSK-type neuro-fuzzy networks (Juang & Lin, 1998; Lin & Xu, 2006; Takagi & Sugeno, 1985). For Mamdani-type

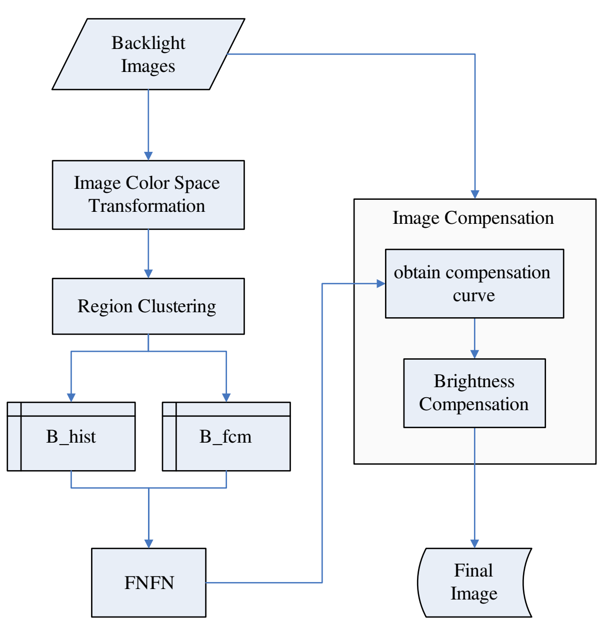


Fig. 1. Flow chart of the proposed compensation method.

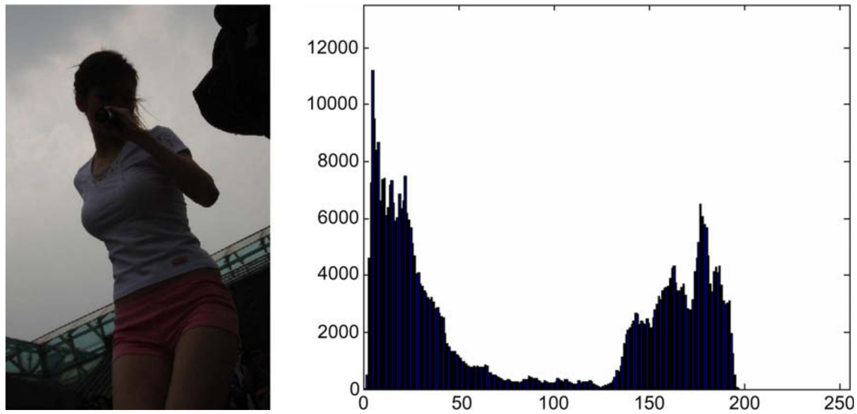


Fig. 2. The example of the backlight image and histogram.

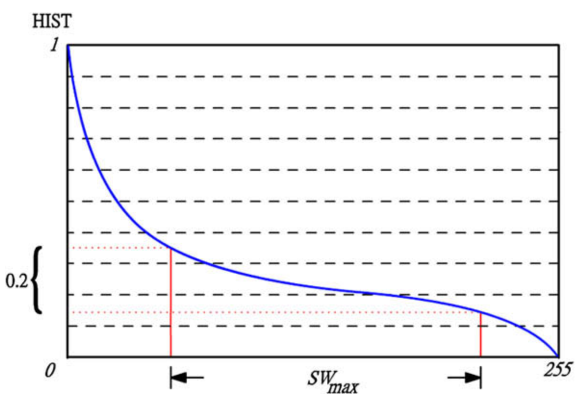


Fig. 3. The HIST histogram and the max window.

neuro-fuzzy networks, the minimum fuzzy implication is used in fuzzy reasoning. Meanwhile, for TSK-type neuro-fuzzy networks, the consequence of each rule is a function input variable. The general adopted function is a linear combination of input variables plus a constant term. Many researchers (Juang & Lin, 1998; Lin & Xu, 2006) have shown that using a TSK-type neuro-fuzzy network achieves a superior performance in network size and learning accuracy than that of Mamdani-type neuro-fuzzy networks. In the classic TSK-type neuro-fuzzy network, which is a linear polynomial of the input variables, the system output is approximated locally by the rule hyper-planes. Nevertheless, the traditional TSK-type neuro-fuzzy network does not take a full advantage of the mapping capabilities that the consequent part might offer. The TSK-type model result didn't make the efficient satisfaction in some non-linear or high complexity problems. Therefore, this article uses a more valid network structure, and the consequent part of the rules has led to the FLNN models (Patra & Pal, 1995; Patra et al., 1999), is named the functional-link-based neuro-fuzzy network (FNFN). Each fuzzy rule that corresponds to a FLNN consist the functional expansion of the input variables. The orthogonal polynomials and linearly independent functions are adopted as functional link neural network bases.

This subsection describes the FNFN model, which uses a nonlinear combination of input variables. Each fuzzy rule corresponds to a sub-FLNN, comprising a functional link. Fig. 4 presents the structure of the proposed FNFN model. Nodes in layer 1 are input nodes, which represent input variables. Nodes in layer 2 are called membership function nodes and act as membership functions, which express the input fuzzy linguistic variables. Nodes in this layer are adopted to determine Gaussian membership values. Each node in layer 3 is called a rule node. Nodes in layer 3 are equal to the number of fuzzy sets that correspond to each external linguistic input variable. Links before layer 3 represent the preconditions of the rules, and links after layer 3 represent the consequences of the rule nodes. Nodes in layer 4 are called consequent nodes, each of which is a nonlinear combination of the input variables. The node in layer 5 is called the output node; it is recommended by layers 3 and 4, and acts as a defuzzifier.

The FNFN model realizes a fuzzy if-then rule in the following form:

$$\text{Rule-}j: \text{IF } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j} \ldots \text{and} x_i \text{ is } A_{ij} \ldots \text{ and } x_N \text{ is } A_{Nj}$$

$$\begin{aligned} \text{THEN } \hat{y}_j &= \sum_{k=1}^{M} w_{kj}\phi_k \\ &= w_{1j}\phi_1 + w_{2j}\phi_2 + \cdots + w_{Mj}\phi_M, \end{aligned} \tag{6}$$

where x_i and $\hat{y}_j$ are the input and local output variables, respectively; A_{ij} is the linguistic term of the precondition part with Gaussian membership function; N is the number of input variables; w_{kj} is the link weight of the local output; φ_k is the basis trigonometric function of the input variables; M is the number of basis function, and Rule-j is the jth fuzzy rule.

The operation functions of the nodes in each layer of the FNFN model are now described. In the following description, $u^{(l)}$ denotes the output of a node in the lth layer.

Layer 1 (input node): No computation is performed in this layer. Each node in this layer is an input node, which corresponds to one input variable, and only transmits input values to the next layer directly:

$$u_i^{(1)} = x_i. \tag{7}$$

Layer 2 (membership function node): Nodes in this layer correspond to a single linguistic label of the input variables in Layer 1. Therefore, the calculated membership value specifies the degree to which an input value belongs to a fuzzy set in layer 2. The implemented Gaussian membership function in layer 2 is

$$u_{ij}^{(2)} = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right), \tag{8}$$

where m_{ij} and σ_{ij} are the mean and variance of the Gaussian membership function, respectively, of the jth term of the ith input variable x_i.

Layer 3 (rule node): Nodes in this layer represent the preconditioned part of a fuzzy logic rule. They receive one-dimensional membership degrees of the associated rule from the nodes of a set in layer 2. Here, the product operator described above is

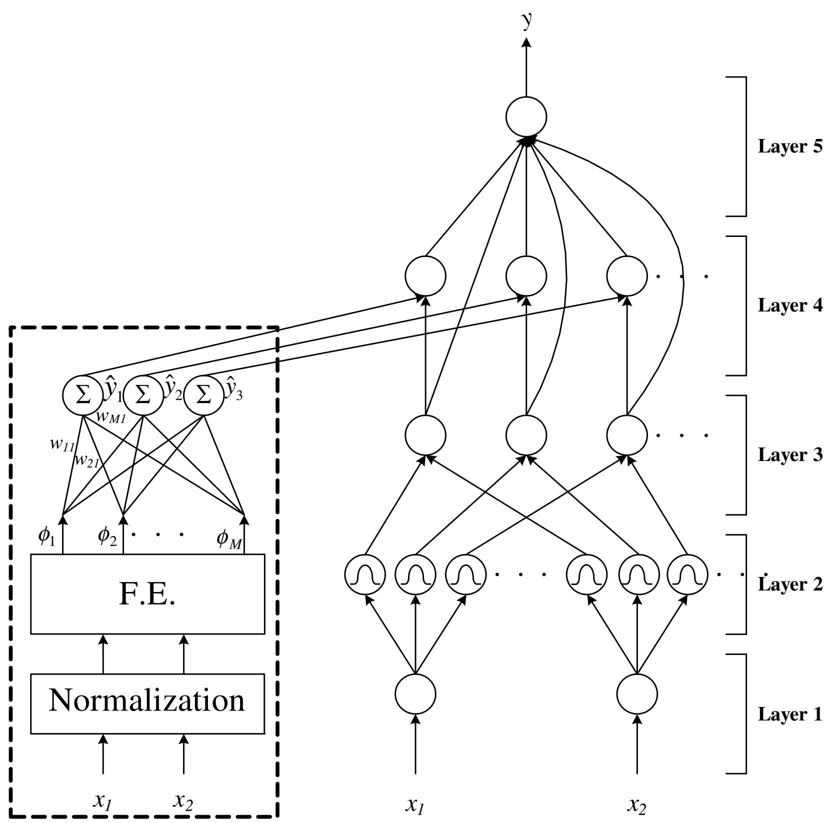


Fig. 4. Structure of proposed FNFN model.

adopted to perform the IF-condition matching of the fuzzy rules. As a result, the output function of each inference node is

$$u_j^{(3)} = \prod_i u_{ij}^{(2)}, \tag{9}$$

where the $\prod_i u_{ij}^{(2)}$ of a rule node represents the firing strength of its corresponding rule.

Layer 4 (consequent node): Nodes in this layer are called consequent nodes. The input to a node in layer 4 is the output from layer 3, and the other inputs are nonlinear combinations of input variables from a functional link neural network, where the nonlinear combination function has not used the function tan*h(.)*, as shown in Fig. 4. For such a node,

$$u_j^{(4)} = u_j^{(3)} \cdot \sum_{k=1}^{M} w_{kj}\phi_k, \tag{10}$$

where w_{kj} is the corresponding link weight of functional link neural network and ψ_k is the functional expansion of input variables. The functional expansion uses a trigonometric polynomial basis function, given by $[x_1 \sin(\pi x_1) \cos(\pi x_1) x_2 \sin(\pi x_2) \cos\ (\pi x_2)]$ for two-dimensional input variables. Therefore, M is the number of basis functions, $M = 3 \times N$, where N is the number of input variables.

Layer 5 (output node): Each node in this layer corresponds to a single output variable. The node integrates all of the actions recommended by layers 3 and 4 and acts as a defuzzifier with,

$$y = u^{(5)} = \frac{\sum_{j=1}^{R} u_j^{(4)}}{\sum_{j=1}^{R} u_j^{(3)}} = \frac{\sum_{j=1}^{R} u_j^{(3)} \left(\sum_{k=1}^{M} w_{kj}\phi_k\right)}{\sum_{j=1}^{R} u_j^{(3)}} = \frac{\sum_{j=1}^{R} u_j^{(3)} \hat{y}_j}{\sum_{j=1}^{R} u_j^{(3)}}, \tag{11}$$

where R is the number of fuzzy rules, and y is the output of the FNFN model.

As described above, the number of tuning parameters for the FNFN model is known to be $5 \times N \times R$, where N and R denote the number of inputs and existing rules, respectively.

2.4. The proposed immune particle swarm optimization learning algorithm

In neuro-fuzzy networks, we need to use some learning algorithm for network parameter adjusting. Many neuro-fuzzy networks with the learning of the network parameter were done by using backpropagation (BP) learning algorithm (Jishuang, Chao, & Zhengzhi, 2003; Juang & Lin, 1998; Lin & Lin, 1997; Nauck & Kruse, 1993), but it is based on gradient descents that are easily trapped at local minima. The other drawback is aiming at a different network, we must describe it by different mathematical models, and it will increase the complexity of solving the problem. Recently, evolutionary computation has designed to do the optimization of parameters for neuro-fuzzy networks and it solves optimization problems. GA (Farag, Quintana, & Germano, 1998; Lin & Xu, 2006) and IA (Kalinli & Karabogab, 2005; Wen & Song, 2004; Zuo, Li, & Ban, 2003) are two popular evolutionary algorithms that simulate the biological behavior and human physiological function. Many researches have already been successful to utilize these two kinds of evolutionary algorithms to solve a lot of problems. GA and IA are very efficient at exploring the global search space, but the problems about local minimum and premature convergence still exist. Therefore, for the sake of enhancing the search ability of the global best solution and avoiding trap in a local optimal solution, we proposed an improved algorithm which combines PSO and IA, is named the immune particle swam optimization (IPSO) to realize network parameter learning in FNFN. The PSO (Kennedy & Eberhart, 1995), was proposed by Kennedy and Eberhart, has proved to be very effective for solving global optimization according to a small amount of calculation, and it is very easy to understand and implement. In order to avoiding the trapping in a local optimal solution and to ensuring the searching capability of near global optimal solution, mutation plays an important role in IPSO.

Therefore, we employ the merits of PSO to improve mutation mechanism of immune algorithm.

This subsection describes an efficient immune particle swarm optimization (IPSO) learning method for the FNFN model design. Analogous to the biological immune system, the proposed algorithm has the capability of seeking feasible solutions while maintaining diversity. The proposed IPSO is combining the immune algorithm (IA) and the particle swarm optimization (PSO) to perform parameter learning. The IA uses the clonal selection principle to accelerate the search and increase global search capacity. The PSO algorithm has been proved to be very effective for solving global optimization. It is not only a recently invented high-performance optimizer that is very easy to understand and implement, but it also requires little computational bookkeeping and generally only a few lines of code. In order to avoid trapping in a local optimal solution and to ensure the search capability of a near global optimal solution, mutation plays an important role in IPSO. Therefore we employed the advantages of PSO to improve the mutation mechanism of immune algorithm. A detailed IPSO is presented in Fig. 5. The whole learning process is described step-by-step below.

2.4.1. Coding step

The coding scheme consists of the coding done by the IPSO. The IPSO codes the adjustable parameters of a FNFN into an antibody, as shown in Fig. 6, where MS_i represents the parameters of the antecedent of the ith rule in the FNFN and C_i represents the parameters of the consequent of the ith rule, respectively. In this paper, a Gaussian membership function is used with variables representing the mean and deviation of the membership function. Each fuzzy rule in Fig. 4 has the form in Eq. (6), where m_{ij} and σ_{ij} represent a Gaussian membership function with mean and deviation of the jth dimension and ith rule node and w_{im} represents the corresponding parameters of consequent part, and m is equal the M in Eq. (6).

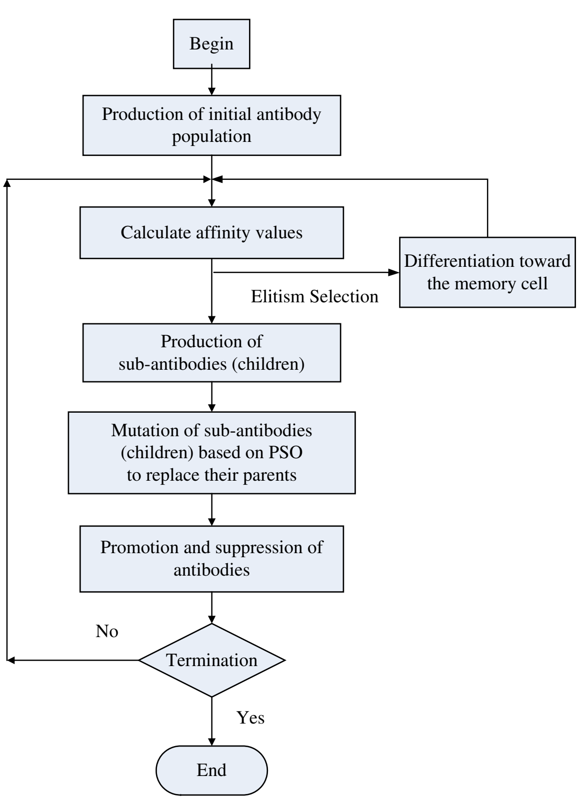


Fig. 5. Flowchart of the proposed IPSO.

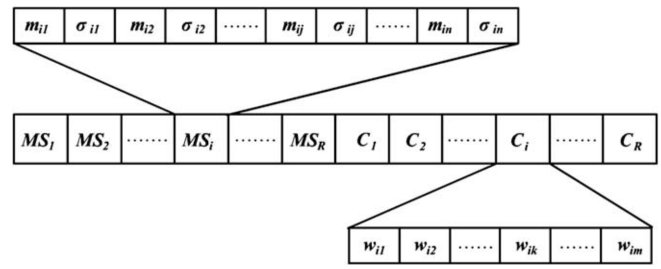


Fig. 6. Coding a FNFN into an antibody in the IPSO method.

2.4.2. Initial population production

In the immune system, the antibodies are produced in order to cope with the antigens. In other words, the antigens are recognized by a few of high affinity antibodies. All of the initial antibodies utilizing a real variable string are generated by random.

2.4.3. Calculate affinity values

For the large number of various antigens, the immune system has to recognize them for their posterior influence. In biological immune system, affinity refers to the binding strength between a single antigenic determinants and an individual antibody-combining site. The process of recognizing antigens is to search for antibodies with the maximum affinity with antigens. In this paper, the affinity value is designed according to the follow formulation:

$$\text{Affinity value} = \frac{1}{\sqrt{\frac{1}{N_t}\sum_{k=1}^{N_t}(y_k - y_k^d)^2}}, \tag{12}$$

where y_k represents the kth model output, y_k^d represents the desired output, and N_t represents the number of the training data. In the problems, the higher affinity refers to the better antibody.

2.4.4. Production of sub-antibodies

In this step, we will generate several neighborhoods to maintain solution variation. This strategy can prevent the search process from becoming premature. We can generate several clones for each antibody on feasible space by Eqs. (13) and (14). Each antibody regards as parent while the clones regard as children (sub-antibodies). In other words, children can regard as several neighborhoods of near parent.

$$\text{mean and deviation} : \text{clons}[\text{children}_{i\_c}] = \text{antibody}[\text{parent}_i] + \alpha, \tag{13}$$

$$\text{weight} : \text{clons}[\text{children}_{i\_c}] = \text{antibody}[\text{parent}_i] + \beta, \tag{14}$$

where $parent_i$ represents the *ith* antibody from the antibody population; $children_{i\_c}$ represents clones number c from the *i-th* antibody; α and β are parameters that control the distance between parent.

2.4.5. Mutation of sub-antibodies based on particle swarm optimization

In order to avoid trapping in a local optimal solution and to ensure the search capability of near global optimal solution, mutation plays an important role in IPSO. Through the mutation step, only one best child can survive to replace its parent and enter the next generation. Hence, we employed the advantages of particle swarm optimization (PSO) to improve mutation mechanism.

PSO is not only a recently invented high-performance optimizer that is very easy to understand and implement, but it also requires little computational bookkeeping and, generally, only a few lines of code. Each particle has a velocity factor $\rightharpoonup v_i$ and a position factor $\rightharpoonup x_i$ to represent a possible solution. The velocity for each particle is updated by

$$\vec{v}_i(k+1) = \omega * \vec{v}_i(k) + \phi_1 * \text{rand}() * (\text{Lbest} - \vec{x}_i(k)) + \phi_2 * \text{rand}() * (\text{Gbest} - \vec{x}_i(k)), \tag{15}$$

where ω is the coefficient of inertia, φ_1 is the cognitive study, and φ_1 is the group study. The rand() is uniformly distributed random numbers in [0,1]. The term $\vec{v}_j$ is limited to the range $\pm \vec{v}_{\max}$. If the velocity violates this limit, it will be set at its proper limit. Changing velocity enables every particle to search around its individual best position and global best position. Based on the updated velocities, each particle changes its position according to the following:

$$\vec{x}_i(k+1) = \vec{x}_i(k) + \vec{v}_i(k+1). \tag{16}$$

When every particle is updated, the affinity value of each particle is calculated again. If the affinity value of the new particle is higher than those of local best, then the local best will be replaced with the new particle. The mutation step flowchart is presented in Fig. 7.

2.4.6. Promotion and suppression of antibodies

In order to affect antigens and keep diversity to a certain degree, we use information entropy theory to measure the diversity of antibodies. If the affinity between two antibodies is greater than the suppression threshold Th_{aff}, these two antibodies are similar, and the antibody of lower affinity value is reduced a small amount of value λ. The antibodies with high antigenic affinity are transformed into long-lived B memory cells; others with low antigenic affinity are affected. Fig. 8 shows the immune algorithm composed of N antibodies having L genes.

From information entropy theory, we get

$$IE_l(N) = \sum_{i=1}^{N} -P_{il} \log P_{il}, \tag{17}$$

where P_{il} is the probability that the *i-th* allele comes out at the *l-th* gene. The diversity of the genes is calculated using Eq. (13). The average entropy value *IE(N)* of diversity can be also computed as follows:

$$IE(N) = \frac{1}{L} \sum_{l=1}^{L} IE_l(N), \tag{18}$$

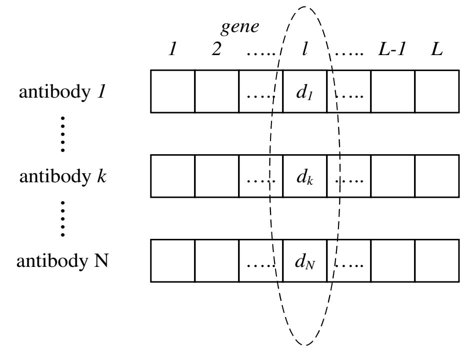


Fig. 8. The coding of antibody population.

where L is the size of the gene in a antibody. There are two kinds of affinities in IPSO. One explains the relationship between an antibody and an antigen using Eq. (12). The other accounts for the degree of association between the *j-th* antibody and the *k-th* antibody and measures how similar these two antibodies are. It can calculated by using

$$\text{Affinity\_}Ab_{jk} = \frac{1}{1 + IE(2)}. \tag{19}$$

2.4.7. Elitism selection

When a new generation is created, the risk of losing the best individuals is always existent. In this study, we adopt elitism selection to overcome the above-mentioned problem. Therefore, the antibodies are ranked in ascending order to their affinity value. The best individuals are kept as the parent for the next generation. Elitism selection improves the efficient of IPSO considerably, as it prevents losing the best result.

2.5. Image compensation

In the previous sub-section, we proposed a procedure to detect the backlight degree of an image. After the backlight degree is detected, we can compensate the backlight image according to the

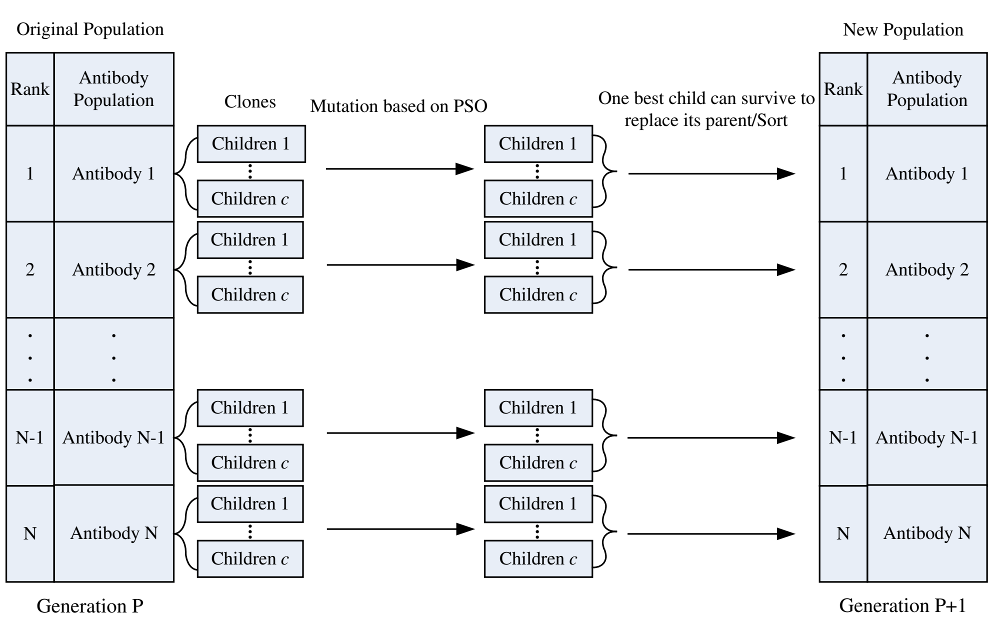


Fig. 7. The flowchart of the mutation step.

detected backlight degree. In light of the characteristic of the contrast degree in backlight image, we have to increase the luminance of backlight object and reduce the luminance of background. Therefore, this paper depends on the adaptive cubic curve method which is determined by the backlight degree to compensation a backlight image. The element that the curve constitutes includes the upward and downward parabolic curves, and the curve can be adjustable by tuning point (*TP*). The curve shown in the following figure: Fig. 9.

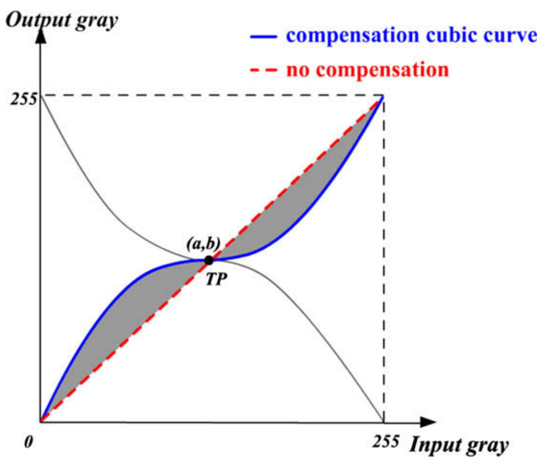


Fig. 9. An adaptive cubic curve.

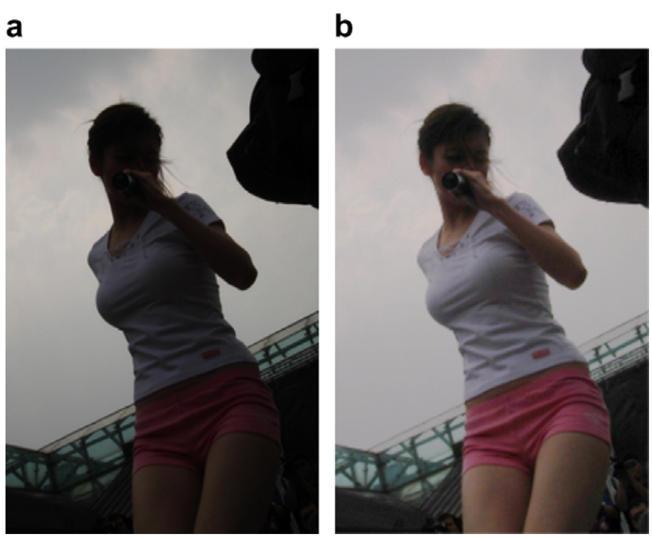


Fig. 10. (a) The original image and (b) the compensated image.

Table 1
The initial parameters before training

Parameters	Value
Antibody population Size	50
Coding type	Real number
Clones number c	5
ω	0.25
φ_1	0.8
φ_2	1.25
Suppression threshold (Th_{aff})	0.8

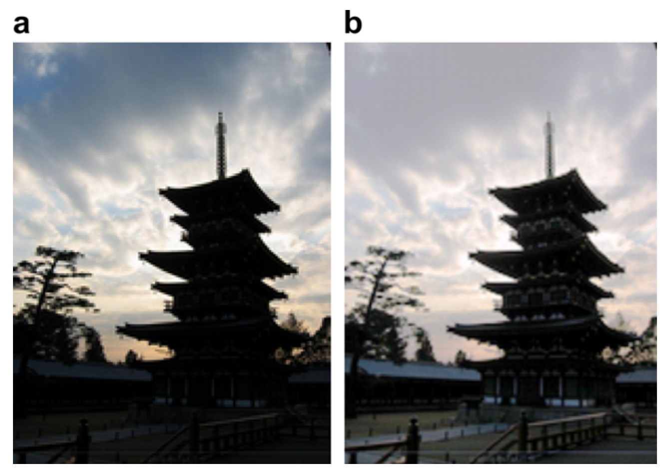


Fig. 12. (a) The original image and (b) the compensated image.

The tuning point *(a,b)* is determine by backlight degree which is computed through a well-trained FNFN. The cubic curve which is described by Eq. (20):

$$f(x) = \begin{cases} \frac{-b}{a^2}(x-a)^2 + b, & \text{if } x < a, \\ \frac{(255-b)}{(255-a)^2}(x-a)^2 + b, & \text{otherwise}. \end{cases} \tag{20}$$

The above-mentioned equation is combined with the half-upward curve and the half-downward parabolic curves, and the value a and b is calculated as:

$$\begin{cases} a = (C_1 + C_2)/2 \\ b = a + (B_{\text{degree}} \times (C_{2-a})) \end{cases}, \tag{21}$$

where C_1 and C_2 are the cluster center which are obtained by FCM algorithm; B_{degree} is the backlight degree which is obtained by FNFN. Finally, we can get a compensated image, when all pixels in backlight image are transformed via the adaptive cubic curve.

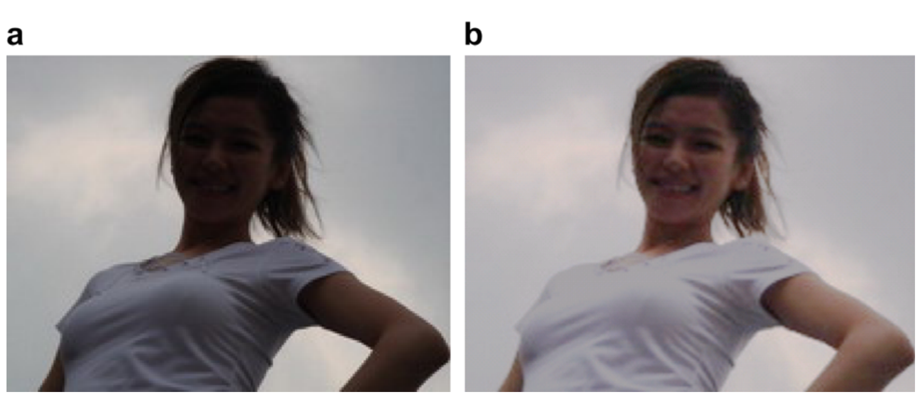
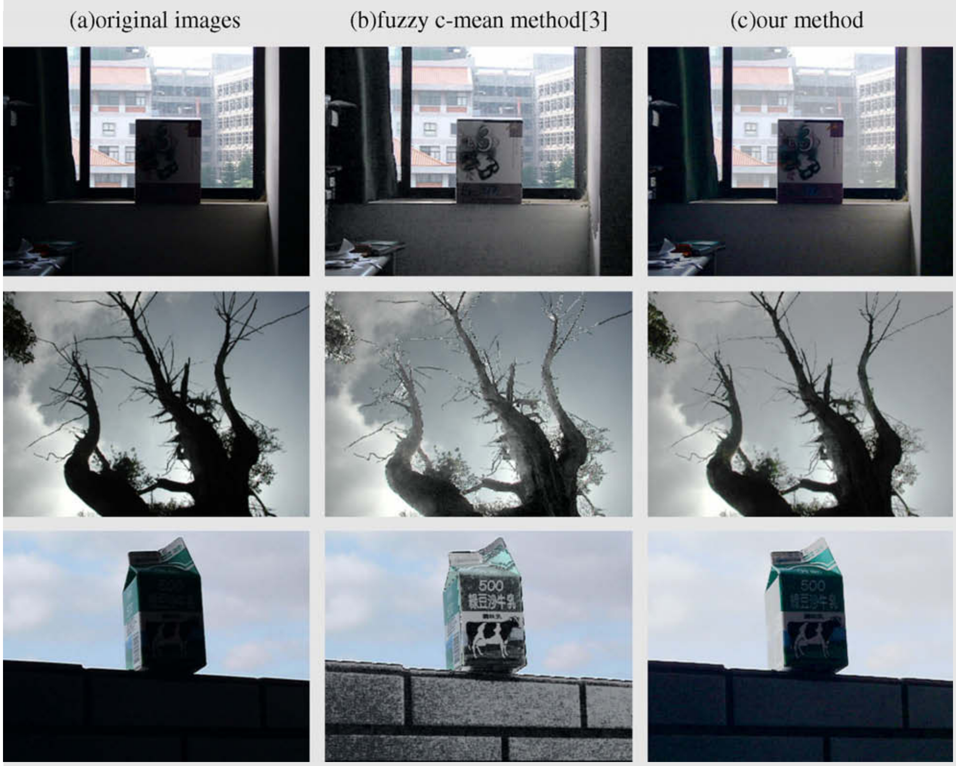


Fig. 11. (a) The original image and (b) the compensated image.

Table 2
Performance comparison of various methods



3. Experiment results

In this section, we will apply the proposed FNFN with IPSO learning to estimate the backlight degree of a backlight image. The initial parameters of FNFN with IPOS are shown in Table 1.

We compensated three backlight images for testing the efficiency of the proposed method. Figs. 10(a), 11(a) and 12(a) show the original backlight images while Figs. 10(b), 11(b) and 12(b) show the compensated images.

In addition, we also compared our method with that of other method (Chin & Lin, 2005). In Chin & Lin (2005), they compensated backlight images with fuzzy C-mean learning algorithm and fuzzy inference. The comparison results are shown in Table 2. In this table, the first column is the original backlight images. The second column is the compensated images access by fuzzy C-mean method (Chin & Lin, 2005) and the third column is the compensated images access by our method. In Chin & Lin (2005), a two-stage processing technique utilizing the fuzzy c-means learning mechanism and the fuzzy logic rule inference is proposed to compensate the backlight images. In their method, the luminance is only adjusted in the local area (i.e., the object) of the backlight images. Therefore, the edge the local area of the backlight images is too sharp. For this reason, we use an adaptive cubic curve for compensating the whole area of each backlight images. In Table 2, we can find that the edges between object and background of our method are more natural than fuzzy c-mean method.

4. Conclusions

In this paper, a new backlight image compensation method, which combines the proposed FNFN with IPSO learning method and the adaptive multi-cubic curve, is proposed. In the backlight level detecting stage, we have extracted two operative backlight factors, and the FNFN model is based on the two factors that can accurately detect the compensation degree. In the compensation stage, we using adaptive compensation curve effectively to improve image backlight problem and to accord with the generalization state of backlight images. Simulation results show that our method can obtain more natural image backlight compensation in the edges between object and background.

References

Chen, C. H., Lin, C. T., & Lin, C. J. (2007). A functional-link-based fuzzy neural network for temperature control 2007. In *Proceedings of the IEEE symposium on foundations of computational intelligence (FOCI 2007)* (pp. 53–58).

Chin, C. L., & Lin, C. T. (2005). Detection and compensation algorithm for backlight images with fuzzy logic and adaptive compensation curve. *International Journal of Pattern Recognition and Artificial Intelligence, 19*(8), 1041–1057.

Farag, W. A., Quintana, V. H., & Germano, L. T. (1998). A genetic-based neural fuzzy approach for modeling and control of dynamical systems. *IEEE Transactions on Neural Networks, 9*(5), 756–767.

Halgamuge, Saman K. (1998). A trainable transparent universal approximator for defuzzification in Mamdani-Type neural fuzzy controllers. *IEEE Transactions on Fuzzy Systems, 6*(2).

Jishuang, Q., Chao, W., & Zhengzhi, W. (2003). Structure-context based fuzzy neural network approach for automatic target detection. In *Proceedings of the IEEE international, geoscience and remote sensing symposium, IGARSS'03* (Vol. 2, pp. 767–769).

Juang, C. F., & Lin, C. T. (1998). An on-line self-constructing neural fuzzy inference network and its application. *IEEE Transactions on Fuzzy Systems, 6*(1), 12–31.

Kalinli, A., & Karabogab, N. (2005). Artificial immune algorithm for IIR filter design. *Engineering Applications of Artificial Intelligence, 18*, 919–929.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).

Lee, C. Y., Lin, C. J., & Chuang, H. C. (2006). Efficient immune-based particle swarm optimization learning for TSK-Type neuro-fuzzy network design. In *Proceedings of the 17th workshop on object-oriented technology and applications*.

Lin, D. T., & Huang, C. C. (2003). Image back-light compensation with fuzzy C-Means learning algorithm and fuzzy inferencing. *ISSPA2003*, Paris, France (Vol. 1, pp. 433–436).

Lin, C. J., & Lin, C. T. (1997). An ART-based fuzzy adaptive learning control network. *IEEE Transactions on Fuzzy Systems, 5*(4), 477–496.

Lin, C. J., & Xu, Y. J. (2006). A hybrid evolutionary learning algorithm for TSK-type Fuzzy model design. *Mathematical and Computer Modelling, 43*, 563–581.

Lin, C. J., & Xu, Y. J. (2006). A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications. *Fuzzy Sets and Systems, 157*(8), 1036–1056.

Nauck, Detlef & Kruse, Rudolf (1993). A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation. In *Proceedings of the IEEE international conference on neural networks ICNN'93* (Vol. 2, pp. 1022–1027).

Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy C-means model. *IEEE Transactions on Fuzzy Systems, 3*(3).

Patra, Jagdish C., & Pal, Ranendra N. (1995). A functional link artificial neural network for adaptive channel equalization. *Signal Processing, 43*, 181–195.

Patra, J. C., Pal, R. N., Chatterji, B. N., & Panda, G. (1999). Identification of nonlinear dynamic systems using functional link artificial neural networks. *IEEE Transactions on System, Man, and Cybernetics, 29*(2), 254–262.

Shimizu, S., Kondo, T., Kohashi, T., Tsumta, M., & Komuro, T. (1992). A new algorithm for exposure control based on fuzzy logic for video cameras. *IEEE Transactions on Consumer Electronics, 38*(3), 617–623.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-15*, 116–132.

Wen, X., & Song, A. (2004). An immune evolutionary algorithm for sphericity error evaluation. *International Journal of Machine Tools & Manufacture, 44*, 1077–1084.

Zuo, X. Q., Li, S. Y., & Ban, X.J. (2003). An immunity-based optimization algorithm for tuning neural fuzzy controller. In *Proceedings of the second international conference on machine learning and cybernetics Xi'an* (Vol. 2, pp. 666–671).