An Improved Genetic Algorithm for Fast Face Detection Using Neural Network as Classifier

Masanori Sugisaka[1,2], and Xinjian Fan[1]

[1]Department of Electrical and Electronic Engineering, Oita University, Oita, Japan

[2]The Institute of Physical and Chemical Research (RIKEN) at Nagoya, Anagahora, Shimoshidami, Moriyama-ku Nagoya, 463-0003, Japan

(Tel : +81-97-554-7831; E-mail: {msugi, fxinjian}@cc.oita-u.ac.jp)

Abstract: This paper presents a novel method to speed up neural network (NN) based face detection systems. NN-based face detection can be viewed as a classification and search problem. The proposed method formulates the search problem as an integer nonlinear optimization problem (INLP) and develops an improved genetic algorithm (IGA) to solve it. Each individual in the IGA represents a subwindow in an input image. The subwindows are evaluated by how well they match a NN-based face filter. A face is indicated when the filter response of the best particle is above a given threshold. Experimental results show that the proposed method leads to a speedup of 83 on 320×240 images compared to the traditional exhaustive search method.

Keywords: Genetic algorithm, evolutionary computation, face detection, INLP, neural network

1. INTRODUCTION

Fast and robust face detection is an important computer vision problem with applications to surveillance, multimedia processing, and HCI. Face detection is often formulated as a classification and search problem: a search strategy generates potential image regions and a classifier (filter) determines whether or not they contain a face. A standard approach is exhaustive search, in which the image is scanned in raster order and every $n \times n$ window of pixels over multiple image scales is classified [1].

Neural networks have been proven to be a powerful tool to discriminate between face and non-face patterns when trained a large number of examples. So far the most accurate detection performance has been obtained by using neural network-based methods [2, 3]. However, these methods are generally computationally expensive because: (a) the search window is a high dimensional vector that has to be classified in a very non-linear space; (b) there are hundreds of thousands of windows to search.

Although many efforts have been done to reduce the runtime of neural network based methods, most of them focused on reducing the computational complexity of classifiers such as using PCA to reduce the dimensionality of the input vector [4], using FFT to calculate neural activities efficiently [5], etc. Only a few attentions were given to improving the search efficiency. In Ref. [6], the search window moves every q pixels (q=3~5) instead of every pixel. Thus the number of searched windows is only about $1/q^2$ of the exhaustive search, but with the disadvantage of lowering the system's performance. Many methods use skin color information to limit the search area [6, 7]. But color information is not always able to be used and it is very difficult to build a skin color model robust to illumination changes.

In this paper, to reduce computational cost while retaining high detection accuracy, we propose a new search method for neural network (NN) based face detection systems. The method is based on the idea that the face search (FS) problem can be formulated into an integer nonlinear optimization problem (INLP). The integer variables are parameters that represent a subwindow in an input image. The objective function is based on the output of a face filter.

Genetic algorithms (GAs) are adaptive optimization techniques that simulate the mechanics of genetic evolution of creatures and have been shown to perform well in large search spaces. Inspired by their mechanism, in this paper we use GAs to solve the FS problem formulated as an INLP. However, it is known that the simple genetic algorithm (SGA) has the drawback that it is easy to fall into premature convergence, which makes it perform poorly for difficult problems such as presented here. In this paper, we proposed an improved genetic algorithm (IGA) for the formulated INLP. Experiments show that the IGA is more efficient for our problem than the SGA.

Based on a NN-based face filter, this paper presents an IGA for the FS problem formulated as an INLP. The feasibility of the proposed method is demonstrated and compared with the exhaustive search method on a set of 42 test images with promising results. In this paper, we assume that there is only one face contained in the test image. The extension of the method to detect multiple faces will be done in our future work.

2. FORMULATION OF FS AS AN INLP

Since a learned face filter should response strongly near the face position while its output on the background should be low, we can locate a face by finding a local maximum filter response which value is above a threshold. Thus the face search (FS) problem can be formulated as an optimization problem:

Let **T** represent an input image, **SW** represent a subwindow and dv be its detection value (the corresponding output of the neural network). With these notations the FS problem can be stated as:

$$\arg\max_{\mathbf{SW}} dv(\mathbf{SW}) \qquad \forall \mathbf{SW} \in \mathbf{T} \tag{1}$$

If

$$dv^* \geq threshold \tag{2}$$

The corresponding portion of **SW** is declared as a face, where dv^* is the best detection value found so far and *threshold* the given threshold value of neural network output.

Because the state variables that represent a subwindow in an image only take integer values, the formulated optimization problem is in fact an integer nonlinear optimization problem (INLP).

3. NEURAL NETWORK BASED FACE FILTER

The purpose of the face filter is to classify a window of size

20×20 pixels extracted from an image, as a face or as a non-face.

We use a retinally connected neural network [3] to serve as the face filter. The network takes a 20×20 pixel window as input. Each hidden unit receives inputs only from part of the input layer (called a *receptive field*). There are 3 kinds of receptive fields: four 10×10 pixel regions, sixteen 5×5 pixel regions, and six 20×5 pixel overlapping horizontal stripes. Each of these receptive fields has full connection to two hidden neurons. It has a single output. The output is a real value from -1.0 to 1.0, giving the likelihood as to what extent the input window looks like a face.

The neural filter was trained using standard back-propagation. The face training set is composed of 1000 frontal faces (positive examples). Each face image was normalized into 20×20 pixels. Fifteen additional face examples were generated from each original face image by randomly rotating it (up to 10º), scaling (90% to 110%), translating (up to half a pixel), and mirroring. 9000 random patches chosen from images containing no faces serve as the initial non-face training set (negative examples). Additional non-faces were introduced by applying the bootstrap algorithm[3]. Both the face and non-face examples were enhanced by the preprocessing procedures as described in Subsection 5.2.

4. AN IMPROVED GENETIC ALGORITHM

Genetic algorithms (GAs) are adaptive optimization techniques that simulate the mechanics of genetic evolution of creatures and have been shown to perform well in large search spaces. However, the premature convergence of GAs has been noted. Furthemore, although the rate of convergence is very fast during the early stages of the algorithms, a drastic reduction in convergence velocity in the latter generations is often encountered before GAs provides an accurate solution. This is an illustration of one of the disadvantages of genetic algorithms [8].

In this paper, a new genetic algorithm is proposed. The new GA is resulted by adding an *individual similarity checking* (ISC) module into the simple GA (Figure 1). The module first

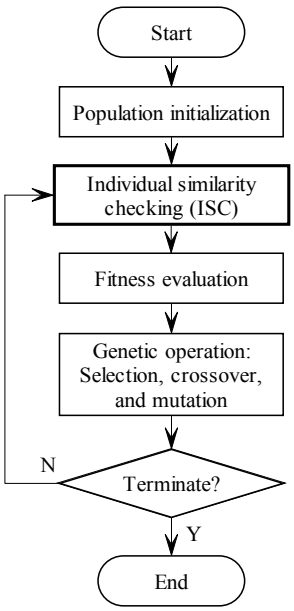


Fig. 1 Flowchart of the IGA

```
for i=1 to N do:
  for j=1 to i-1 do:
    if d_ij<R_c then do:
      Replace individual i with a randomly
      generated new invidual
    endif
  endfor
endfor
```

Fig. 2 Pseudo code of the ISC module

calculates the similarity scores between two individuals in the population. If the two individuals are highly similar, one of the individuals will be replaced by a randomly generated new individual. Thus the diversity of the population is improved. The module works as shown in Figure 2, where N is the population size, d_{ij} is the distance between two individuals, i and j, and R_c the similarity threshold. d_{ij} is defined as:

$$d_{ij} = \sum_{n=1}^{M} K_n \times \frac{\left| x_n^{(i)} - x_n^{(j)} \right|}{b_n - a_n} \tag{3}$$

where M is the number of individual parameters, $x_n^{(i)}$ and $x_n^{(j)}$ are the nth parameters of individuals i and j respectively, K_i is the similarity weight of the nth parameter, and b_n and a_n are the upper and lower limits of the nth parameter space.

5. FACE SEARCH USING IGA

The main steps of the proposed method are shown in Figure 3. In the following, we will describe the approach in detail.

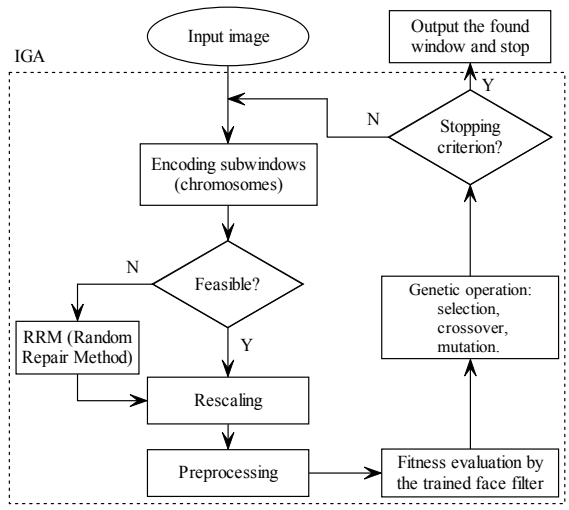


Fig. 3 Main steps of the proposed method

5.1 Encoding and rescaling

In our problem, each particle represents a subwindow in the input image. We use its center (C_x, C_y) and length S to encode a subwindow. To evaluate subwindows of different sizes using the neural network, we should rescale them to the size of 20×20 (the input size of the neural network). However, if this computation is done on every size of subwindows, it will be very time-consuming. To avoid it, we first build an image

pyramid†:

$$W \times H,\ \frac{W}{q} \times \frac{H}{q},\ \cdots,\ \frac{W}{q^k} \times \frac{H}{q^k},\ \cdots,\ \frac{W}{q^L} \times \frac{H}{q^L} \tag{4}$$

where W and H are the width and height of the input image respectively, and q is the scale factor. The top level (level L) should have a size more than 20×20:

$$\frac{\min(W, H)}{q^L} \geq 20 \text{ , gives}$$

$$L = \left\lfloor \frac{\ln(\min(W, H)) - \ln 20}{\ln q} \right\rfloor \tag{5}$$

Then we let S to be chosen among the following geometric sequence†:

$$20,\ 20q,\ \cdots,\ 20q^k\ \cdots,\ 20q^L \tag{6}$$

For a subwindow $\mathbf{SW} = (C_x, C_y, \lfloor 20q^k \rfloor)^T$, we find its mapped 20×20 window $\mathbf{SW}' = (C_x^{'}, C_y^{'}, 20)^T$ in level k of the pyramid by:

$$C_x^{'} = \left\lfloor \frac{C_x}{q^k} \right\rfloor,\ C_y^{'} = \left\lfloor \frac{C_y}{q^k} \right\rfloor \tag{7}$$

So each particle $\mathbf{X}$ is constructed as $\mathbf{X} = (C_x, C_y, k)^T$. C_x, C_y and k are defined in [10, W-10], [10, H-10] and [0, L] respectively.

5.2 Preprocessing

Before a 20×20 window is passed to the trained neural network, it is preprocessed with lighting correction (by subtracting a best fit linear function) and histogram equalization as in Ref. [2, 3]. The former reduces the effect of different lighting conditions and the latter improves contrast across the window.

5.3 Fitness evaluation

To evaluate each particle (subwindow), we directly use its detection value (the corresponding output of the neural filter): the larger its detection value (dv), the more the subwindow resembles a face. The fitness function $f(\mathbf{SW})$ is given as

$$f(\mathbf{SW}) = dv(\mathbf{SW}) \quad \mathbf{SW} \in \mathbf{T} \tag{8}$$

where $\mathbf{T}$ is the input image and $\mathbf{SW}$ is a subwindow, $dv(\mathbf{SW}) \in [-1,\ 1]$.

The corresponding subwindow of a particle may go beyond the image's boundary even if all its variables lie in the search boundary. To guarantee feasibility of solutions, we investigated a random repair method (RRM): if a particle is checked to be infeasible, it will be forced to "fly" to a new, randomly generated position. The method works as follows:

If $\mathbf{SW} \notin \mathbf{T}$, then

Step 1: Randomly generate a new position $\mathbf{SW}'$.

Step 2: If $\mathbf{SW}' \in \mathbf{T}$, replace $\mathbf{SW}$ with $\mathbf{SW}'$; otherwise, go to step 1.

In EAs, the classical approach to deal with infeasible solutions is to add a penalty term to the fitness function [10]. The proposed RRM has proven more efficient for our problem than the penalty approach.

5.4 Genetic operation

Based on their fitness, individuals in the population are guided by three genetic operators to possible face regions in the image. New (C_x, C_y, k) generated by GA are real values. When corresponding to a subwindow in the input image, they are transformed into integers by using the floor function. During flying, if a variable extends the defined search boundary, it will be set to the closest limit, i.e.

$$x_j = \begin{cases} x_{j\min} & \text{if } x_j < x_{j\min} \\ x_{j\max} & \text{if } x_j > x_{j\max} \end{cases} \tag{9}$$

where x_{jmin} and x_{jmax} are respectively the lower and upper search limit of variable x_j, $x_j \in \mathbf{X}$.

5.5 Stop criterion

The algorithm is stopped when 1) a "face" is found – the detection value of the best individual is above the given threshold or 2) the maximum iteration number is reached.

6. EXPERIMENTS

A number of experiments were performed to evaluate the proposed method. The experiments were performed on 42 images with complex backgrounds. Some of the images were chosen from CMU Test Set [11] and other Internet resources; the others were taken by us in an indoor environment using a CCD camera. Each image contains only one face and all the faces can be detected by the neural filter. All the images have the same size of 320×240 and the face size ranges from 34×34 to 178×178.

According to pre-simulation, the parameters of the IGA (real-coded) were set as:

Table 1 Settings for the IGA

Crossover operator	BLX-α crossover [9]
Mutation operator	Gaussian mutation [9]
Selection	Linear rank selection with elitism
Size of population	100
Maximum generation	100
Probability of crossover	0.90
Probability of mutation	0.10

6.1 Experiments

For each image in the test set, we ran our algorithm 100 times. The total detection results are listed in Table 2. Some examples are shown in Figure 4. The threshold of the neural network output was set to 0.1. The time consuming was reported on an AMD Athlon 750 MHz PC with Windows 2000 as its OS.

As shown in Table 2, the proposed search method yielded a high success rate (93.8%) on average (the best is 100% and the worst is 56%). Moreover, about 37% of the failures are because the IGA fell into false detections, and the other failures are due to non-convergence. A further reduction of false detections can be achieved by arbitrating among multiple networks [3]. From the examples shown in Figure 4, we can see that the proposed method maintains robustness in images which contain faces under a very wide range of conditions including scale, pose, position, backgrounds, illumination conditions, etc.

Table 3 gives the comparison of the proposed search method (we call it a *genetic search* method) with the exhaustive search method. It's clear that the time consuming

† Each term in Equ. (4) and (6) is transformed from a real value to an integer value by using the floor function.

Table 2 Experimental results

Success	False	Non-convergence	ANESs	ATC (ms)
93.8%	2.29%	3.91%	1959	242

False: false detection rate
ANESs: Average Number of Evaluated Subwindows
ATC: Average Time Consuming to find a face

Table 3 Genetic search vs. exhaustive search

	Genetic search	Exhaustive search	Ratio
ANESs	1959	193737	1 : 99
ATC (ms)	242	20169	1 : 83

and the number of subwindow evaluations of the proposed method are much less than those of the exhaustive search. Although with a little loss of detection rate (due to non-convergence), a great speedup has been achieved by using the swarm search compared to using the exhaustive search.

6.2 Comparison with search with the SGA

Figure 5 shows the performance comparison of the proposed IGA with SGA. It can be seen that the IGA outperforms SGA both in success rate and in number of needed subwindow evaluations. As shown in Figure 5, although the success rate of SGA increases quickly in the initial iterations, then it spends most of its time making little progress due to premature convergence. It also shows that the proposed ISC is an effective strategy to prevent premature convergence by avoiding two individuals too closely.

6.3 Comparison with other speedup methods

Table 4 shows the comparison of the proposed method with other NN-based face detection methods in processing time. As introduced in Section 1, these methods also take some measures to reduce the detection time. Although the comparison is not accurate for the different systems were tested on different computers and using different sizes of images, we think our system is faster.

7. CONCLUSION

This paper presents a new search method for NN-based face detection. The proposed method formulates the problem of face search into an integer nonlinear optimization problem (INLP) and develops an improved genetic algorithm (IGA) to solve it. The feasibility of the proposed method is demonstrated on a set of 42 images with promising results.

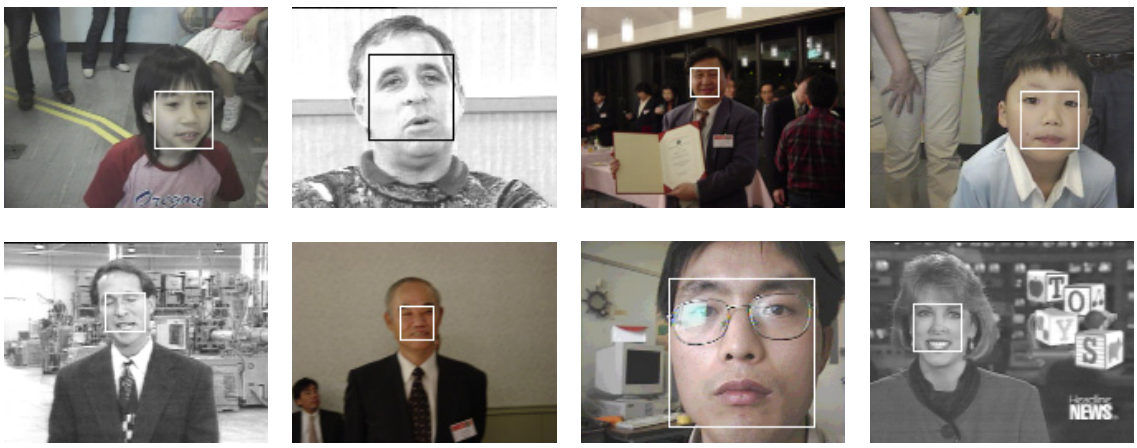
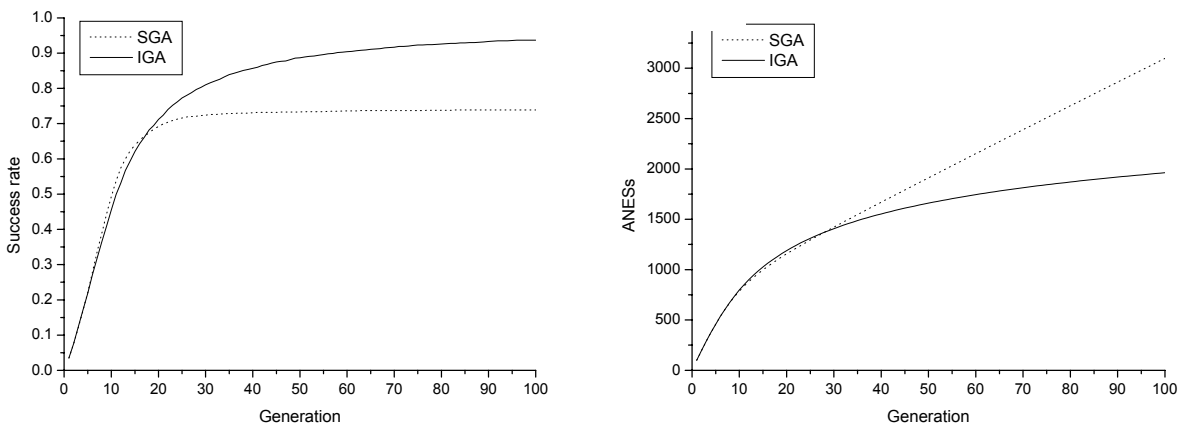


Fig. 4 Examples from the test set



(a) Success rate vs. generation (b) ANESs vs. generation

Fig. 5 Performance plots of SGA and IGA

Table 4 Comparison with other NN-based face detection methods

	Swarm search	Rowley [3] fast	Huang [4]	Fasel [5]	Feraud [6]
Image size	320×240	320×240	320×240	192×144	108×108 ~ 1024×1024
Computer	AMD Athlon 750 MHz	175 MHz R1000 SGI O2 workstation	Pentium 990 MHz	Sun UltraSparc 30 workstation	DEC Alpha 333
Processing time (second/image)	0.242	2	15	0.7	2.9 (average)

With fine-adjusted parameters, the IGA only requires less than 2000 evaluations of subwindows for finding the face in an image. The results are much more effective and superior over the classical exhaustive search method. Many object detection problems can be formulated as an INLP and the results indicate the possibility of IGA as a practical tool for various INLPs of object detection.

However, we have found that the method doesn't works well on some images, especially when the face size is very small. And for simplification, only single-face detection is considered in this paper. How to improve the robustness and extend the method to multiple face detection is the future work.

REFERENCES

[1] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. & Mach. Intell.*, Vol. 24, No. 1, pp. 34-58, 2002.

[2] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Anal. & Mach. Intell.*, Vol. 20, No. 1, pp. 39-50, 1998.

[3] H. A. Rowley, "Neural network-based face detection," *Thesis submitted for the degree of Doctor of Philosophy*, School of Computer Science, Carnegie Mellon University, 1999.

[4] L. Huang, A. Shimizu, Y. Hagihara, and H. Kobatake, "Face detection from cluttered images using a polynomial neural network," *Proc. Int'l Conf. on Image Processing*, Vol. 2, pp. 669-672, Thessaloniki, Greece, 2001.

[5] B. Fasel, "Fast multi-scale face detection," *Technical Report COM-98-04*, IDIAP, 1998.

[6] R. Feraud, O. J. Bernier, J. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *IEEE Trans. Pattern Anal. & Mach. Intell.*, Vol. 23, No. 1, pp. 42-53, Jan. 2001.

[7] S. Karungaru, M. Fukumi, and N. Akamatsu, "Human face detection in visual scenes using neural networks," *Trans. of Institute of Electrical Engineers of Japan*, Vol. 122-C, No. 6, pp. 995-1000, 2002.

[8] H. Kitano, "Empirical studies on the speed of convergence of neural network training using genetic algorithms," Proc. AAAI-90, pp. 789-795, 1990.

[9] F. Herrera, M. Lonzano, and J.L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis," Artificial Intelligence Review, Vol. 12, No. 4, 1998.

[10] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," *Proc. 4th Annual Conf. on Evolutionary Programming*, MIT Press, Cambridge, MA, pp. 135-155, 1995.

[11] CMU Test Set: http://vasc.ri.cmu.edu/idb/images/face/frontal_images/images.tar