Genetic Algorithms based Adaptive Search Area Control for Real Time Multiple Face Detection using Neural Networks

STEPHEN KARUNGARU, MINORU FUKUMI,
TAKUYA AKASHI[1] and NORIO AKAMATSU

Department of Information Science and Intelligent Systems,
University of Tokushima,
2-1, Minami-Josanjima, Tokushima 770-8506.
JAPAN.

[1]Department of Electrical and Electronics Engineering,
Yamaguchi University,
2-16-1, Tokiwadai, Ube, Yamguchi, 755-8611.
JAPAN
karunga@is.tokushima-u.ac.jp

Abstract: - Fast and automatic face detection from visual scenes is a vital preprocessing step in many face applications like recognition, authentication, analysis, etc. While detection of a single face can be accomplished with good accuracy, multiple faces detection in real time is more challenging not only because of different face sizes and orientations, but also due to limits of the processing power available. In this paper, we propose a real time multiple face detection method using multiple neural networks and an adaptive search area control method base on genetic algorithms. Although, neural networks and genetic algorithms may not be suitable for real time application because of their long processing times, we show that high detection accuracies and fast speeds can be achieved using small sized effective neural networks and a genetic algorithm with a small population size that requires few generations to converge. The proposed method subdivides the face into several small regions, each connected to an individual neural network. The subdivision guarantees small size networks and presents the ability to learn different face regions features using region-specialized input coding methods. The genetic algorithm is used during the real time search to extract possible face samples from face candidates. The fitness of the face samples is calculated using the neural networks. In the successive frames, the search area is adaptively controlled based on the information inherited from the proceeding frames. To prove the effectiveness of our approach we performed real time simulation using an inexpensive USB camera.

Key-Words: - Adaptive search area control, Genetic Algorithms, Neural networks, Real-time processing, feature extraction.

1 Introduction

Face image based research has been actively pursued by many researchers for a long time and this trend is predicted to continue in the future. This is not at all surprising considering the numerous real world applications that can benefit from real time face detection and recognition. Such applications include surveillance, web and video searching, contrast enhancement for digital cameras, etc. However, a face is a complex object to detect because of the various poses it can take due to its three-dimensional nature. Therefore, because of the infinitely many sizes and orientations that a face can be in at any given time, it is very difficult to train, a prior, a finite system that is capable of detecting faces in all the possible profiles. This makes detection of faces in real time a challenging task. Consequently, to improve the detection accuracy, in addition to finding high quality extractable features, it is necessary that faces are detected fast regardless of their profiles. In offline applications, a neural network system trained with whole frontal face samples produces acceptable detection results [1]. However, when such a system is extended to real time detection, the results are not satisfactory. The reason for the drop in detection accuracy is that, in real time the extractable facial features are constantly changing rendering the offline system that depends of fixed features ineffective. In addition, the system running speed is too slow for online usage. In this work, to partially accommodate the facial changes in real time systems, we propose to subdivide the face into smaller independent sub-regions. It is hoped that, as compared to the use of the whole face as one region, smaller face regions would be more resistant to the changes. Therefore, we concentrate on developing a detection method that learns the small unit regions using small sized fast neural networks. To solve the size and orientation problem, we propose using a genetic algorithm to decide the extraction position, size and orientation of the face samples. In short, we subdivide a face sample into several parts and use smaller specialized neural networks to learn each region.

Sample division also provides the opportunity to learn different face region's features using the best input feature extraction coding for each sub-region. For example, whereas to code the lips regions, the CIE-Yxy color space is considered best due to the redness of the lips [2], it is better to learn other skin color region features using the YIQ color space coding [1]. Moreover, edge features are the best suited for the regions surrounding the eyes [3]. Therefore, sub-dividing the face should, based on these factors, produce better results than the traditional method that use the whole face with a single uniform coding method and a single neural network.

The minimum size of the face detectable by this system, the sample size, is set to 30x30pixels for reason explained later in section 2.

In conventional methods, during the system testing, the samples are usually extracted sequentially from the image, that is, all image positions are tested. Moreover, since the learned sample is of fixed size, image pyramid methods [4] must be used to detect faces larger than the set sample size. However, within the constraints of real time processing, such testing methods are not suitable for our system. Therefore, a genetic algorithm (GA) is introduced for high speed, size and rotation invariance detection of the faces. In the proposed method, instead of sliding the sample through the target image and then scaling the image (pyramid method), the GA is used to extract samples of different sizes and orientations at random positions inside the image. Moreover, to ensure fast search speeds and convergence of the GA, after each run, the new search area is automatically adjusted to the most probable face location based on the proceeding frames detection results. This search method is called adaptive search area control.

The rest of this paper is organized as follows. Sub-section 1.1 reviews the related works by other researchers. In section 2, the design of the subdivided neural networks is detailed. Section 3 explains how high speed search is possible using the genetic algorithm adaptive search area control. Computer simulations and results are in section 4 while section 5 concludes the paper.

1.1 Related works

Currently, many researchers are activley working on face detection, recognition and related areas.

Osadchy et.al [5], presents face detection and pose estimation using energy-based models. Their system can detect face poses for 90^0 in yaw, 45^0 in roll and 60^0 in pitch while Viola et al [6] proposes a robust real time object detector using an "integral image" and other systems. Although they achieve good results, their system is very complex because it involves a combination of a wide range of methods. Wang et. al. [7] propose a hybrid system using sound and vision. In the paper, initial face positions are estimated acoustically from microphone array data. Other works in face detection include use of color and shape information by [8] and view based methods that involve building seperate detectors for different views. The detectors can then be applied in parallel [9],[10],[11],[12] or using a prepocessor to selecet the most suitable detector [13][14].

Off-line, the face detector presented in this paper is an improvement of [1] and achieves an accuracy of 99.3% on the University of Oulu database [15]. It matches the detector in [15], but at the rate of 30 milliseconds per image.

2 Divide and conquer

Neural networks are excellent in generalization and can produce very good classification results for many non-linear, complex and multi-dimensional problems. However, some of their main drawbacks include long training times, large memory requirements especially for large neural networks and lack of set rules to define the optimum number of layers and nodes per layer to solve a particular problem. In this work, in order to take advantage of the neural networks generalization power while reducing the effects of the above mentioned problems, we propose to use a "divide and conquer" principle to solve the online face detection problem. In [1], one large neural network (NN) was used to learn the whole face sample at once. Encouraging offline results (over 97% accuracy) were achieved even for the large neural network. Apart from the long training times and slow running of the large neural network, the other major disadvantage of the method was that, by using the whole face image as a sample, valuable time was wasted learning face regions that did not contribute to the overall system accuracy. One example of such regions was the bottom corners of the face sample (Note that due to the face shape, these regions will usually be part of the background). This was confirmed by a careful examination of the trained weights. The weights in the said regions had very near zero values. This means that their contribution to the final result was very minimal.

Therefore, to overcome the waste of valuable time training the neural network that includes unnecessary weights, we experimented with learning selected parts of the face by subdividing the face into sub-areas of interest. This is the basis of the "divide and conquer" method. Another major motivation for this system design is that, by subdividing the face into small regions, we can then extract the best feature for each region and using the best input feature extraction coding for it, train the neural network

The experience gained from working with large neural networks, for example the one in [1], can confirm that the important features in the face that should be learned include the eyes, eyebrows, nose and lips. Therefore, these areas should definitely be included as face subsections. Other subsections can also be learned, but at a different priority level. Therefore, there should be a minimum of four small neural networks; one each for the eye and eyebrow regions, the nose and the lips.

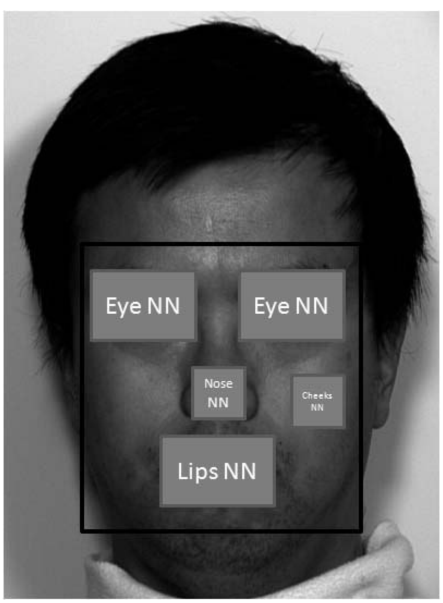

Fig. 1. The sub-regions used to train the neural networks for real time face detection.

If required, skin color regions on the cheeks can also be learned using another neural network. The details of the specialized small neural network are detailed below. Fig. 1 shows the five sub-regions selected as neural network inputs.

The minimum detectable sample size of 30x30 pixels was selected because it was the minimum size at which the facial features are still extractable in real time images without distortion, using our inexpensive USB camera (BWC-130H01/SV (Buffalo Inc.)). Obviously, larger face samples are better but they require more training and testing times in addition to large memory space for storage.

All the neural networks used in this work are three layered, Fig. 2, and were all trained using the back propagation algorithm [10]. The number of input nodes depends on the size of the sub-image and the input coding method used. The number of nodes in the hidden layer is determined experimentally and all the neural networks have one output node.

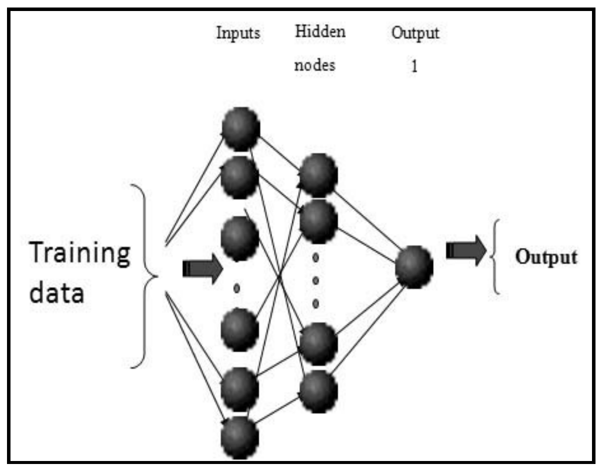

Fig.2 General neural network structure used in the proposed method.

In this work, the size of a neural network is calculated as follows.

Let the sub-image width and height be w and h respectively. Also let each pixel be represented using x components. If the neural network has three layers with y nodes on the hidden layer and z nodes in the output layer, then the size of the neural network can be calculated using eq. (1);

$$I_{size} = \left((w \times h) + 1\right)xy + \left((y + 1)z\right) \qquad (1)$$

Note that the size of the neural network is the total number of weights making up the neural network.

As an example, consider a neural network with a 30x30 pixels input sample. If two components per pixel are used, the neural network input layer with has 1800 nodes. For one hidden layer with 20 nodes and the output layer with one node, then using eq. 1, the neural network size is 36041 weights.

In the following section the small size neural networks (NN) designed in this work are described in details.

2.1 Eye/Eyebrows NN

The eyes and eyebrows regions contain useful features that should be exploited in face detection algorithms, that is, the edges. The decision to combine the eye and eyebrows in the same sub-image is based on the difficulty of separating the two, especially in images taken using an USB camera, at distances of more than two meters.

The size chosen for the sub-section is 15x10pixels. Please note that, this sub-section is among the darkest regions in a face. This is clear from the skin color detection result that fails completely to detect any skin pixels inside the sub-region resulting in dark holes. Fortunately, edge extraction in this area produces very good edges, Fig.3. Therefore, the best feature to chosen for this region is the edge. The canny edge detection filter [17] is used to detect the edges inside the region.

Consequently, each pixel in the input layer is coded using one brightness (edge information) component. Fig. 3 shows the results of the canny edge detector on a given face. Notice that the clear edges extracted around the eyes, eyebrows and nose.

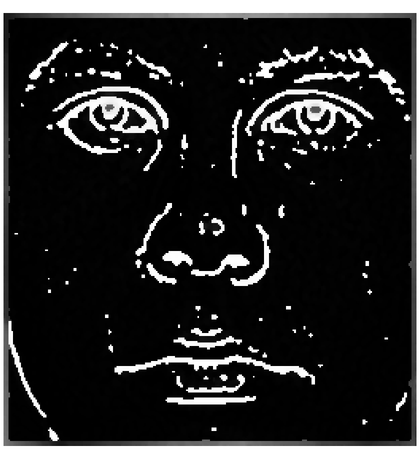

Fig 3. Canny edge detector extraction results

Through experimentation, the hidden layer nodes were set to 8. The output layer contains one node. Therefore, the size of this neural network is 1217 weights. Note that neural networks for the left and right eyes/eyebrow pairs are similar.

2.2 Lips NN

The human lips, regardless of race or place of origin, possess a different color from the rest of the face. This color is red. The lips redness is a common feature in the entire human race. The reason is that the lips region is composed of non-keratinized squamous epithelium that covers numerous capillaries, which give the lips its characteristic color [2], [18], and [19]. In this work, also note that, brightness has little effect on the lips color because the redness is expressed independent of the brightness using the CIE-Yxy color space [19]. For these reasons, we focus on the lips redness as a main feature during detection.

The lips region's sub-image size is set to 15x10pixels, the same as for the eye/eyebrows sub-image. Each pixel is represented by only one component, that is, the X-component from the CIE-Yxy color space. The number of nodes in the hidden layer is set to 10. The size of this neural network is therefore 1521 weights.

2.3 Nose NN

The nose is the face region most affected by the lighting due to its convex shape. Normally, the tip of the nose appears the brightest in a given face image. Therefore, coding the features in this region requires a combination of color and edges information. The color space used is YIQ [20] and the edges are detected using the canny edge detection filter.

The color is represented using the average of the I and Q color components. Therefore each pixel is represented by two components, the average color and the edges information.

The sub-image size is 10x10pixels and there are 10 pixels in the hidden layer of the neural network. Therefore, the size of this neural network is 2021 weights.

2.4 Cheeks NN

Skin color regions make up most of the face. It is therefore, important to learn the feature information it contains. A good sample of the face skin color can be extracted from the cheeks region. This cheeks neural network can learn any useful information contained in the skin color.

The checks sub-image is 10x10pixels. Each pixel is coded using two components from the YIQ color space. The first component is Y representing brightness and the second is the average of the *I* and *Q* color components. With a hidden layer having 10 nodes, this neural network has the same size as the nose NN.

2.5 Size Difference

The neural network in our earlier works using one image had 36041 weights.

By subdividing the image and using five small neural networks, the combined network size becomes 7997 weights. The difference is 28044 weights, a 77.9% reduction in size.

In addition, the time it takes to train the neural networks is expected to be considerably less for the small neural networks.

2.6 NN Training Data

Based on the experience gained from our earlier neural network experiments [1] it can be concluded that for offline usage, training a neural network using “normally available” images work well. By “normally available” we mean scanned face images from photos and newspapers, pictures taken by a digital camera, downloaded from the Internet, etc. However, when tested online, a neural network trained using such images failed to produce acceptable results. On investigation, it was found out that the images used to train the neural network are of considerable higher quality than those encountered online.

Therefore, the images used to train the NNs in this work were collected using the USB camera. 400 such face images were used to train the NNs. Initially, 200 non-face images were collected for training. During training, more non-face images were corrected using the backstrap algorithm [16].

3. Adaptive Search Area Control

After the neural networks are fully trained, testing on visual scenes can be done using several different methods. One such method is running the neural network pixel by pixel throughout the image. That is, for each pixel position in the image, extract a face sample and test it using the trained neural network. This can only detect faces that are the same size as the training samples. To detect faces larger than the learned size, the image must be re-sampled as in an image pyramid. The re-sampling rate must be chosen such that no faces are missed. Therefore, the total time taken to search a given image for faces depends on its original size, the size after every re-sampling and the time taken to run the neural network per each position. For example consider the following case.

- Image size (I_{size}) = 10x10 pixels
- Re-sampling rate (r) = 0.9
- Number of re-samples (ns) = 1 time
- Time taken by NN per pixel (tt) = 0.001secs.

For this case, the total time taken is given by:

1. The time taken to search the original image:

$$I_{size} \times tt = 100 \text{x} 0.001 = 0.1\text{secs.}$$

2. Time take to search each re-sampled image;

$$ns \times I_{size} \times I_{size} \times tt =$$
$$1 \times 100 \times 0.9 \times 0.001 =$$
$$0.09\text{secs}$$

3. Therefore, the total time = 0.19secs.

Another method involves searching for the faces in smaller image regions known as face candidates [12]. Face candidates regions are extracted from the image based on factors like skin color. However, inside the face candidates, the face must still be searched for pixel by pixel using the method described above. The time taken can be calculated as shown above. Since face candidates are smaller regions compared to the original image, the search speed improves. However, the improvement in speed is still slow for online usage with neural networks.

Consequently in this work, inside the face candidates, genetic algorithms (GA) with adaptive search area control are proposed to speed up the face detection speed to levels applicable online.

3.1 Face Candidate Regions

Face candidate regions are defined as skin color regions because it is assumed that faces must contain some skin color in them. In the YIQ color space, skin color regions can be extracted using simple thresholds because skin color occupies a well define region in the color histogram based on the color space. In the color space, by careful selection of a minimum and maximum color thresholds enclosing skin color, it is possible to isolate the skin color regions in an image.

To select the suitable thresholds, skin color data was collected using the USB camera and from it, a color histogram using the Y component verses the sum of the I and Q components plotted. A single threshold derived from the sum of the two color components was then derived, eq. 2.

$$7 \le SumIQ \le 48 \qquad (2)$$

Where SumIQ is the sum of I and Q components in the YIQ color space.

The skin color pixels segmented out using eq. 2 are then grouped together based on their locations. The smallest rectangle surrounding each region becames an independent face candidates. Since only one face is searched for in a face candidate, then the maximum number of faces that can be detected is equal to the number of face candidates extracted.

3.2 GA Structure

Genetic algorithms optimization can be used inside the face candidates to perform a more efficient search by combining all the procedures of the convectional pixel by pixel search method into one process. However, for GAs to speed up the search, their processes of selection and reproduction must converge fast. Usually, this requires a small population, few generations and a small search space. Moreover, trade off is usually necessary because the parameters are usually in competition.

The parameters that the GA must optimize are; the face location in the image, its size and its orientation.

The GA was designed as follows. Each individual has six genes. They represents the horizontal and vertical face sample position (P_x, P_y), horizontal and vertical translation values (T_x, T_y) (used by the GA to vary the sample position), scaling factor (*Scl*) and orientation angle (*Ang*) respectively. These parameters were decided based on the total search area and the size of the original sample.

In this work, the GA was employed inside the face candidates regions. The face candidates regions are skin color regions extracted using a skin color detector. The GA's parameters are decided as follows.

i. The face sample position (P_x, P_y) must be inside the face candidate at all times. The translation values (T_x, T_y) are small random values such that,

$$\le (Tx, Ty) \le x \qquad (3)$$

Where $x = 4$,

Therefore,

$$_{l} \le (Px + Tx) \le W_{xr} \qquad (4)$$

and

$$_{l} \le (Px + Tx) \le H_{yr} \qquad (5)$$

Where, W_{xl} *and* W_{xr} *represent the left and right x-coordinates of the face candidate. Similarly,* H_{yl} *and* H_{yr} *represent the top and bottom y-coordinates of the face candidate.*

This ensures that, $(P_x + T_x)$ and $(P_y + T_y)$ are always inside the face candidate region.

ii. In this work, the maximum clockwise and anticlockwise orientations are set to 30 degrees.

iii. The maximum allowable scale for any face candidate region can be found by dividing the face candidate region's width by the training sample's width.

iv. To further ensure that the sample extracted is within the image, the maximum x-position after scaling and rotation must not be greater than image width minus the sample width. Similarly, the y-position can be found using the image and sample heights.

That makes six genes in the GA's chromosome. Note that, this is a real coded genetic algorithm. Fig.4 shows the chromosome's Structure.

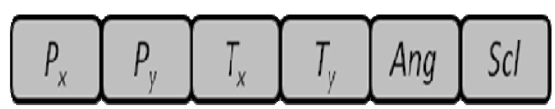

Fig.4. Chromosome Structure

The selection method used is the tournament method. The choice of the tournament selection method is based on the review of the results of the neural networks. From experience, the output range of the neural network is very wide, meaning that the output of non face samples can be very low. For fast convergence, it is better to eliminate such samples during selection. The tournament method is a simple such method.

The initial population is set to 10, taking into account real time usage, the crossover point is selected at random between position 2 and 5 in the chromosome, with the probability of mutation set to 0.0001. The GA is terminated after 10 generations.

To combine the GA and the neural networks, the fitness of each GA sample is calculated from the outputs of the individual neural networks output for the sample. Therefore, the fitness is a weighted average of all the outputs calculated as follows.

$$Fitness = \frac{(O_{el}+O_{er}+O_{lp})+\alpha(O_{ck}+}{5} \qquad (6)$$

Where, O_{el}, O_{er}, O_{lp}, O_{ck} and O_{ns} are the outputs of the left eye, right eye, lips, checks and nose neural networks respectively. α is the priority factor defined as $0\leq\alpha\leq1$. The selection of α determines how much the O_{ck} and O_{ns} networks contribute to the final fitness. In this work, $\alpha = 0.3$.

During reproduction, we save the elite solution, and then use the top 6 of the fittest individuals to reproduce 8 individuals in the next population. The remaining 2 individuals in the next population are reproduced by selection of one of the parents from the top 4 group and the other from the remainder of the population. This method improves the search space by ensuring that we not only retain the best individuals for reproduction but also explore the rest of the population for other possible candidates. However, the worst sample is never selected.

3.3 Fast Speeds: Adaptive Search Area

The GA search method described above works efficiently for offline images and online images with at most two faces present. The search is performed inside the face candidate regions. If the search area in the next frame remains constant, then the search takes about the same time as the earlier frame. As the number of faces candidates increase, there is need to further improve the search speed. This can be achieved by adaptively controlling the search area by inheriting the information from the earlier frames to control the search area. This is accomplished as follows.

i. The system is run on the first frame inside the face candidates.

ii. In the next frame, select the top six chromosomes from the earlier frames and from their positional data, find the minimum and maximum x and y positions.

iii. In the next frame, inherit these values and use them to define a new face candidate region. Note that this region will be less in size than the original face candidate region.

The power of inheriting data from frame to frame can be demonstrated by the improvement in search speed achieved in this method. This simple method produces a very adaptive search area control that ensures that the GA converges even though the number of generations is only ten.

Moreover, all other information contained in the top six chromosomes is inherited into the next frame and used in its GA's initialization. This aids the convergence of the GA because we are starting with better chromosomes in the succeeding frames.

4 Results

After fully training the neural networks and the implementation of the adaptive search area control algorithm, experiments were carried out to prove the effectiveness of our approach.

In this initial experiment, the system was tested under controlled conditions. The conditions were controlled to ensure constant lighting, a clear background and a fixed camera (non camera shaking).

4.1 Experiment Data and Conditions

The experiments were carried out using printed faces images of eight people pasted on the wall of the laboratory.

Three of the faces are hung upright and the other five are pasted at various angles between -5^0 and $+15^0$ respectively. The face rotation is done so as to show that this system is invariant to frontal orientation of ±30 degrees as programmed into the genetic algorithm. The initiall face parameters are shown in Table 1 and 2.

In addition, an image whose color is similar to skin color is also included to show the roboustness of the system in face candidates that do not contain any faces in them.

The invariance of this system to the size of the faces is tested by conducting two experiments with the camera 1 and 2 meters from the faces respectively.

The capture device is the BWC-130H01/SV USB camera from Bufallo Inc. This experiment was carried out on a Dell latitude D400, Pentium M, 1.8GHz, 1GB laptop computer. The image size used in this work is 320x240pixels.

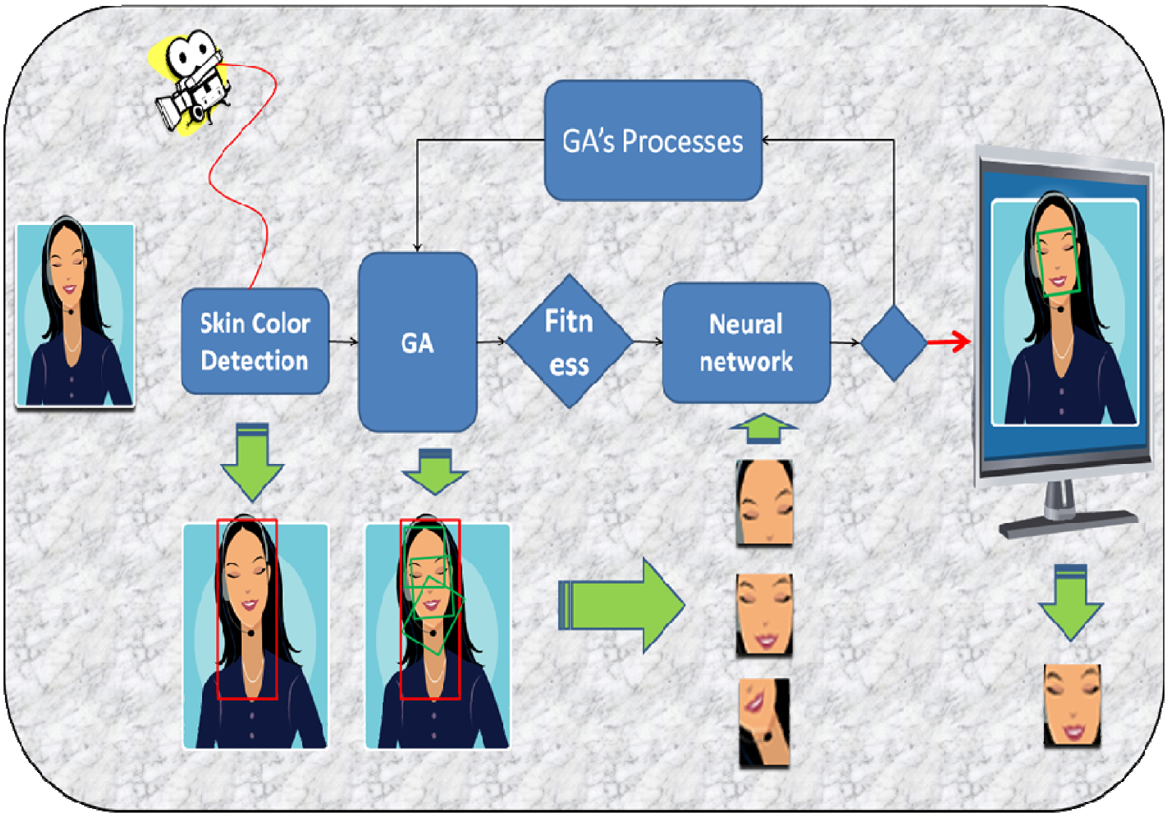

Fig.5. Flow of the proposed algorithm during testing**.**

4.2 Simulation Method

The flow of the proposed system during testing is shown by Fig.5. The system consists of image capture, face candidate extraction and the GA's adaptive search area control processes.

The details of each step, explained using selected images from the results of the experiment conducted, are as follows.

1. After the image is captured, run the skin color detector to extract the skin color regions. These regions are the face candidates, Fig. 6. This continues for five consecutive frames in order to stabilize the face candidates.

2. Run the genetic algorithm inside each face candidate sequentially. The extracted face samples' fitness is individually calculated using the neural networks. Please note that, since the neural networks can only handle fixed size frontal samples, the samples extracted must be scaled and rotated using the parameters derived from the GA.

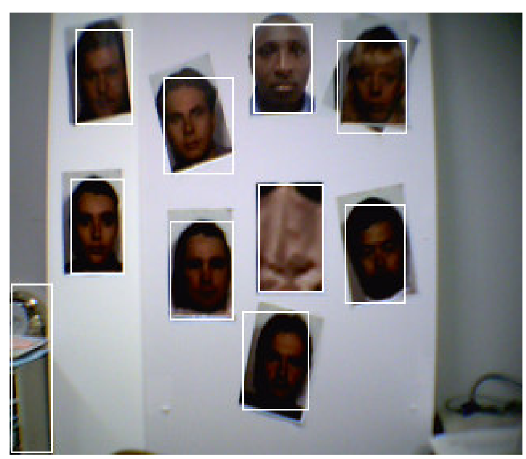

Fig.6. Results of the extraction of the face candidates using the skin color detector.

3. The genetic algorithm processes of selection, reproduction and mutation follow. A sample result is shown in Fig. 7. These processes continue for about 3 frames. The faces detected by this system are classified into two classes. If

the fitness of the elite solution is above 0.95, the face is marked blue, indicating high system confidence. In the second class, solutions between 0.85 and 0.95 are marked pink meaning the system is less confident about the result.

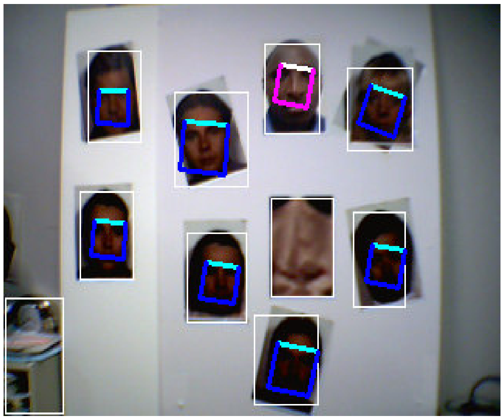

Fig.7. Initial faces locations detected inside the face candidates.

4. After step 3, the rectangle enclosing the fittest six face samples (top left corner coordinates) becomes the new face candidate region. In the proceeding frames, Step 3 is run only once inside the new shrunken face candidates, Fig.8. The new face candidates are the smaller yellow rectangles shown in Fig. 8.

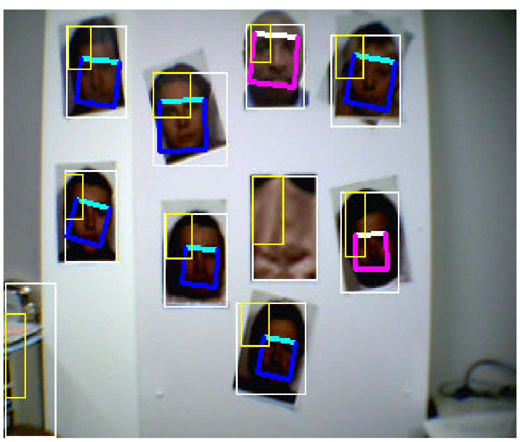

Fig.8. Adaptive search area control. Notice that the original face candidate area has been reduced to less than a quarter of the original size.

5. Steps 1 to 4 are then repeated every second. Therefore, face candidates extraction using the skin color extractor is done for the first five frames per second. After this, the shrunken face candidates are used. Since the skin detector is not run on all the frames the system speed up.

4.3 Experimental Results

In this section, the operating speed and accuracy results of this system are presented.

To evaluate this system, the system was run at 30 frames per second for ten minutes, capturing about 18000 frames. This data was used to calculate the average accuracy and speed of the system.

In this system, of about the 30 frames processed per second, the first five are not searched for faces. Instead, the system uses the five frames to determine the face candidates regions. Therefore, the total frames to be searched for faces are about 14600.

We carried out two sets of experiments, System 1 and System 2.

- **System 1**: The camera is placed two meters from the faces (all the 8 faces are in the camera's view). The face parameters for the system are shown in Table 1.
- **System 2**: The camera is only one meter from the faces (In this case, only five faces are in the camera's view). The face parameters for the system are shown in Table 1.

Table 1 shows the original face parameters for System 1.

Table 1. System 1 Face parameters

Faces	Position		Angle	Scale
	x	y		
1	8	38	5	1.2
2	86	47	15	1.2
3	132	15	2	1.4
4	218	32	-1	1.0
5	64	102	1	1.4
6	122	132	3	1.5
7	198	126	10	1.3
8	165	179	-7	1.3

The original face parameters when the camera is 1 meters from the faces (System 2) are similarly shown in Table 2.

Table 2. System 2 Face parameters

Faces	Position		Angle	Scale
	x	y		
1	90	44	5	1.5
2	142	78	15	1.6
3	210	31	2	1.8
4	74	149	-1	1.3
5	149	190	3	1.6

Table 3 shows the average processing time taken for the main steps of our method. The times shown, however, do not include the time taken to capture and draw the frames. These are the times taken to run our system on the captured frame data. (The real time taken per frame is between 70 and 130 milliseconds). In the table, step 1 is the face candidate extraction, step 2 represents the GA system without search area control and step 3 represents the GA adaptive search control system.

Table 3 Face detection steps times

	Average Time Taken/ Frame (Msec)	System Average (Msec)
Step 1	15	23
Step 2	30	
Step 3	19	

The systems average accuracy and false acceptance rate (FAR) are as shown in Table 4. The FAR represents the number of detections that the system falsely detects as faces. The results are calculated by comparing the face data generated by the GA and the real face data extracted manually beforehand (Table 1 and 2). A below 5% difference in the data is taken as an accurate face extraction.

Table 4 Final System Results

	FAR (%)	Accuracy (%)
System 1	**0.02**	**98.0**
System 2	**1.8**	**97.0**

Fig. 9 shows the visible faces when the camera was brought about 1 meter from the faces. Notice that only five faces are visible.

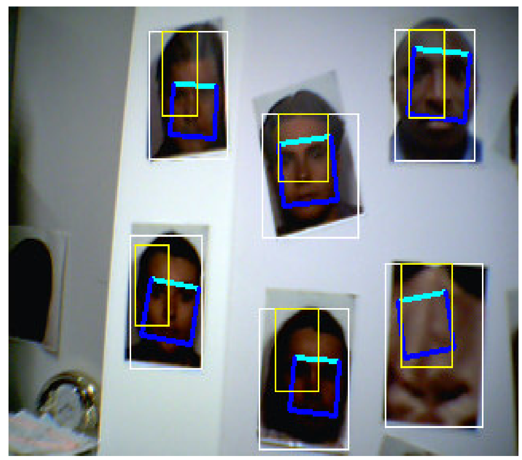

Fig.9. Detection results with the camera 1 meter from the faces. Notice that although all five faces are accurately detected, one false detection results.

4.4 Face Data

Apart from the location of a face on the image, this system also provides information about the face's orientation and size. These parameters can be use by other systems like face recognition to optimize their performance.

The face parameters, that is, position, size and orientation were all found to be within 5% error compared to the real ones (extracted manually). Therefore, it can be concluded that the faces were extracted accurately.

4.5 Selected Results

In this section, we show some results from the System 1's experiment, Fig. 10. These images are the results of 2 seconds of testing. In this system, the system confidence in finding a face is based on the final fitness for every face candidate region. If the fitness is above 0.95, then the face found is marked in blue. If the fitness is between 0.85 and 0.95, then the face is marked pink indicating less confidence. Faces marked in blue are given high reproduction priority in the next generation of the genetic algorithm.

4.6 Discussion

The maximum number of faces that can be detected by this system is equal to the number of face candidates detected because the system assumes that only one face per face candidate region. Since only one face is searched for per face candidate, then in a case where two or more faces are in such an area, only one is likely to be detected and the others will be missed. This situation usually occurs when the faces are very close together or the background causes all faces to be segmented into the same face candidate region. This can highly reduce the accuracy of the system. Therefore, there is great need to improve the face candidate regions detection to reduce the chances of faces being missed.

There is also a difference between the time taken by our system to process a frame and the time taken for the whole capture process. Based on the capture process times, it can be concluded that our system works at 15 and 8 frames per second for systems 1 and 2 respectively. We think that the results of system 2 are lower because of two reasons. One, since the camera is near the faces, the face candidates are very large. Therefore, in addition to the large search space, the sample sizes are likely to be large, straining the system memory. Two, this system is trained with images 30x30pixels in size. In system 2, the face samples are more than 1.5 times this size and we think that this contributes to the lower detection accuracy due to the information lost during the scaling down.

The data used to train the system was acquired with the USB camera about 2 meters from the training faces. This could be the reason the results of System 1 are better that those of System 2. It seems the scaling parameter is not working as well as we had anticipated. We must find ways to make the system more robust to the camera position.

5 Conclusion

We proposed a real time face detection method using several small size neural networks and a genetic algorithm with adaptive search area control. To prove the effectiveness of this system, experiments were conducted. From the experimental results, the final real time face detection average accuracy of 98% was achieved at about 15 frames per second. The system's FAR is about 1.8%. The GA offers size and orientation invariance for the faces and also on the camera's side, some location and orientation invariance. These results are based on ten minutes of simulation. Moreover, the system can provide face parameters like position, size and orientation for use by other application. In addition, the system already transforms the face to zero orientation fontal 30x30pixels images; therefore, the extracted faces are ready for use by other systems, like face recognition.

In the immediate future, the effectiveness of this system in real world environments will be tested including the improvement of the system to be more robust to the camera position. The skin color detection process that determines the face candidates must be improved to ensure that faces are inside the regions. Furthermore, the system must be improved to be able the detection of more than one face per face candidate. We will explore the possibility of introducing a threshold in the GA process such that all the solutions above the threshold can be selected in addition to the elite solution.

Further down the road, the results of this work will be integrated with our robotics works that aims to use a humanoid robot to recognize people. The first challenge will be to effectively run the system on less powerful microprocessors used in robotics.

Fig.10. Two seconds selected results using System 1. The faces are marked in different colors depending on the systems confidence at a given time. Blue represents highest confidence.

References:

[1] S. Karungaru, M. Fukumi and N. Akamatsu: Face Detection: Size and Rotation Invariance using Genetic Algorithms. *Proc. of NCSP2005*, pp. 211-214, 2005.

[2] Takuya Akashi, Yuji Wakasa, Kanya Tanaka, Stephen Githinji Karungaru *and* Minoru Fukumi : High Speed Genetic Lips Detection by Dynamic Search Domain Control, *IEEJ Transactions on Electronics, Infomation and Systems*, Vol.127, No.6, pp.854-866, 2007.

[3] S. Karungaru, M. Fukumi, N. Akamatsu *and* A. Takuya, Automatic Human Faces Morphing Using Genetic Algorithms Based Control Points Selection, *Trans. of International Journal of Innovative Computing, Information and Control*, Vol.3, No.2, pp.247-256, 2007.

[4] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt and J. M. Ogden, Pyramid methods in image processing RCA Engineer, 29-6, pp. 33-41, 1984.

[5] M. Osadchy, Y. LeCun and M. Miller: Synergistic Face Detection and Pose Estimation with Energy-Based Models, *Journal of Machine Learning Research,* 8:1197-1215, 2007.

[6] Paul Viola and Michael J. Jones, Robust Real-Time Face Detection, *International Journal of Computer Vision* 57(2), pp. 137-154, 2004.

[7] C. Wang Brandstein, M.S., A hybrid real-time face tracking system, *proc of IEEE/ICASSP*, Vol. 6, pp. 3737-3740, 1997.

[8] E. Osuna, R. Freund and F, Girosi, Training support vector machines: an application to face detection, *Proc. of the IEEE computer society conference on computer vision and pattern recognition*, pp. 130-136, 1997.

[9] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 84–91, 1994.

[10] K. Sung and T. Poggio. Example-based learning of view-based human face detection. *PAMI*, 20: pp. 39–51, 1998.

[11] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. *Proc. of Computer Vision and Pattern Recognition*, 2000.

[12] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. *Proceedings of the 7th European Conference on Computer Vision, Part IV*, pp. 67-81, 2002.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and pattern Recognition*, pp. 511–518, 2001.

[14] C. Huang, B. Wu, H. Ai, and S. Lao. Omni directional face detection based on real adaboost. *International Conference on Image processing*, vol.1, pp. 593-596, 2004.

[15] Soriano M., Marszalec E., Pietikainen M., Physics-based face database for color research, *Journal of Electronic Imaging*, 9(1) pp. 32-38, 2000**.**

[16] Rowley, Baluja and Kanade, Rotation Invariant neural network based face detection. *Proc. of the IEEE computer society conference on computer vision and pattern recognition*, pp. 38-44, 1998.

[17] J. Canny, A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8, No. 6. 1986.

[18] L. Otis, D. Piao, C. Gibson and Q. Zhu: Quantifying labial blood flow using optical doppler tomography", Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology,98, 2, pp. 189–194 , 2004.

[19] J. H.M. Gloster: "The use of second-intention healing for partial-thickness Mohs defects involving the vermilion and/or mucosal surfaces of the lip", Journal of the American Academy of Dermatology, 47, 6, pp. 893–897, 2002.

[20] K. Plataniotis and A. Venetsanopoulos, *Color image processing and applications*. Springer, Ch.1, 2000.