Multiple Faces Detection in Real Time using Neural Networks

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

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

[1]Department of Electrical and Electronics Engineering,
Yamaguchi University,
2-16-1, Tokiwadai, Ube, Yamguchi, 755-8611.
JAPAN

Abstract: - In this paper, a real time face detection method using several small size neural networks and a genetic algorithm with adaptive search area control is proposed. Neural networks and genetic algorithms may not be suitable for real time application because of their long processing times. However, in this paper, we show how fast speeds can be achieved using small effective neural networks and a genetic algorithm with a small population size that requires few generations to converge. We subdivide the face into several regions, each connected to an individual neural network. This guarantees small size networks and also offers the ability to learn different face regions features using different coding methods. The genetic algorithm is used during the real time search. It extracts possible faces from face candidates that are then tested using the neural networks. The face candidate area is then adaptively reduced depending on the location of the top six face samples. We then performed real time simulation using an inexpensive USB camera to prove the effectiveness of our proposal. We achieved between 98 and 96% accuracy for one or multiple faces respectively at 15 to 8 frames per second.

Key-Words: - Genetic Algorithms, Neural networks, Real-time processing, Adaptive Search Control

1 Introduction

Real time object detection and recognition is a very important research area because numerous real world applications can benefit from it. However, detecting objects, for example faces, in real time is not a trivial task. Faces are three-dimensional objects which appears different when captured in two-dimensions depending on the pose. This makes facial feature extraction difficult. As the number of extractable features decrease, the ability to extract the faces in real time also decreases. In this work, we concentrate on small fast neural networks and fast sample area extraction using adaptive search area control based on a genetic algorithm. Traditionally, one neural network is used to learn a sample. However, in this work, we subdivide the sample to several parts and use smaller specialized neural networks for each region. Sample division also ensures that we can learn different face region's features using the best coding for that sub-region. For example, to code the lips regions, the XYZ color space is considered best due to the redness of the lips. However, it is better to learn other skin color region features using the YIQ color space coding. Moreover, edges features are the best suited for the eyes. Therefore, sub-dividing the face should produce better results than the traditional method. The sample size is set at 30x30pixels for reason explained in section 2. During testing the samples can be extracted sequentially from the image, that is, all image positions must be tested. Moreover, since the learned sample is of fixed size, image pyramid method must be used to detect faces larger than this size. Within the constraints of real time processing, such a testing method is not suitable for our system. Therefore, a genetic algorithm (GA) is introduced for high speed, size and rotation invariance. Instead of sliding the sample through the target image and then scaling the image, the GA is used to extract samples at random positions in the image and of different sizes and orientations. In addition, after each run, the new search area is not the whole image, but an area selected depending on the detection results.

Work related to this include that by Viola et al [1] who propose 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. [2] 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 [3].

Off-line, the face detector presented in this paper is an improvement of [4] and achieves an accuracy of 99.3% on the University of Oulu database [5]. It matches the detector in [6], but at the rate of 30 milliseconds per image.

The rest of this paper is organized as follows. 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. Computer simulations and results are in section 4 while section 5 concludes the paper.

2 Sample Sub-division and Learning

In our earlier work [4], we concentrated on learning the whole sample at once using one neural network (NN). Encouraging results were achieved although the networks were quite large. For example, for a 30x30 pixels sample, using two components per pixel, the neural network input layer has 1800 nodes. If there is one hidden layer with 20 nodes and one output layer node, then the NN size is 36041 weights. Apart from the large neural networks, the other major disadvantage of the method in [4] was learning face regions that do not contribute any useful features, for example, the sample corners.

To overcome these problems, we experimented with learning individual parts using face division. Important features in the face include the eyes, eyebrows, nose and lips. Therefore, these areas should definitely be included as face subsections. The other sections can also be learned, but at a different priority level. Therefore, we should at least have four neural networks, one each for sets of eye and eyebrow (two), the nose and the lips. Skin color regions on the cheeks can also be learned using another neural network. Each of these neural networks is detailed below.

The 30x30 pixels sample size was selected because it was the minimum size at which the facial features are still extractable in real time images. Larger face samples are best but they require more computation time. The neural networks are all three layered and trained using the back propagation algorithm. The number of nodes in the hidden layer is determined experimentally and all the neural networks have one output node.

2.1 Eye/Eyebrows NN

The decision to combine the eye and eyebrows in the same sub-image is base on the difficulty of separating the two, especially in images taken using an USB camera, at more than two meters. The size chosen for the section is 15x10pixels. This section is among the darkest regions in a face. Skin color detection in this area produces very few hits. Therefore, the best feature to use for this region is the edges. The canny edge detection filter [7] is used to detect the edges in the regions. Therefore, each pixel in the input layer is coded using one component.

Through experimentation, the hidden layer nodes were set to 8. The output layer contains one node. The size of this neural network is therefore, 2417 weights. Note that there are similar neural networks for both sides of the face.

2.2 Lips NN

For most of the human race, the lips have a different color from the rest of the face. For most people, the natural color of the lips is reddish. Redness in skin color can best be represented using the XYZ color space. Therefore, this color space is used to code the lips region pixels.

The sub-image size is set at 15x10pixels, the same as for the eye/eyebrows sub-image. Each pixel is represented by two components. One is the X-component of the XYZ color space [8] and the other is the brightness represented by the average of the three XYZ components. With 10 nodes in the hidden layer, this neural network has a size of 6021 weights.

2.3 Nose NN

The nose region is the one most affected by the lighting due to its convex shape. In fact, the tip of the nose will usually appear the brightest in a given face image. Coding this region requires a combination of color and edges information. The color space used is YIQ and the edges are detected using the canny edge detection filter. The color is represented using the average of the I and Q components.

The sub-image size is 10x10pixels with 10 pixels in the hidden layer of the neural network. Therefore, the size of this neural network is 4021 weights.

2.4 Cheeks NN

Skin color regions make up most of the face. It is therefore, important to learn some of the information it contains. This is done using the cheeks neural network. The checks sub-image is 10x10pixels. Each pixel is coded using two components from the YIQ color space. The first is Y for brightness and the second is the I and Q color component average. With a hidden layer having 10 nodes, this neural network has the same size as the nose NN.

2.5 Size Difference

The neural networks in our earlier works using one image had 36041 weights. However, by subdividing the image and using small neural networks, the total network size reduces to about half, that is, 19897 weights. Also, the time it takes to train the neural networks is considerably less for the small neural networks in this work.

2.6 NN Training Data

Experience has shown us that for offline usage, training a NN using normal images work well. By "normal" we mean scanned images, pictures taken by digital camera, etc. However, when such a network is tested online, it fails to produce acceptable results. On investigation, we found out that the images used to train the neural network are of higher quality than those encountered online. Therefore, the images used to train the NNs in this work were collected using an USB camera. 400 such face images were used to train the NNs. Non-face images were collected similarly using the backstrap algorithm.

3 Fast Face Searching

Normally, face searching is conducted 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. To search for 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. To speed up such search methods, face candidates [4] regions are often used. Face candidates regions are extracted from the image based on factors like skin color. However, the face must still be searched for pixel by pixel inside the face candidates. Since face candidates are smaller regions compared to the original image, the search speed improves. However, this improvement is still slow for online usage with neural networks. Consequently, genetic algorithms (GA) are proposed to solve this problem.

The GAs can 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. This would require a small population, few generations and a small search space. Needless to say, these parameters are usually in competition and for best results, trade-off is usually important.

3.1 GA Structure

In our earlier work [4], we developed a search GA as follows. The GA chromosome represents the sample position (P_x,P_y), translation values (T_x,T_y), scale (S) and orientation (A). The parameters were decided based on the search area and the original sample size. This design can be reproduced here as follows. The sample position can be any position within the image taking into account that no area of the sample should be outside the image. Therefore, the maximum x-position should be image width minus sample width. Similarly, the y-position can be found using the image and sample height. The translation values are set to between plus and minus 4 pixels, with the maximum orientation set to plus/minus 30 degrees. The maximum scale can be found by dividing the image width with the sample width. Therefore, there are six genes in this GA. Note that, this is real coded genetic algorithm. Additionally, if face candidates are used, then the GA chromosomes parameters can be calculated by substituting the image with the face candidate region. Table 1 shows the chromosome's genes.

Table 1 Chromosome genes

Parameter	Value
x-position,P_x	Image width –sample width
y-position,*Py*	Image height –sample height
translation values (T_x,T_y),	+/- 4
Max scale (S)	Image width /sample width
Max orientation (A)	+/- 30

The selection method used is the roulette wheel. The initial population is set to 10, taking into account real time usage, the crossover point is set at 3 and the

probability of mutation is 0.0001. The GA is terminated after 10 generations. The fitness of each GA sample is the averaged output of the individual neural networks output.

As in [4], during reproduction, we save the elite solution, and then use the top 40% of the fittest individuals to reproduce 80% of the next population. The remaining 20% of the next population is reproduced by selection of one of the parents from the top 40% 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.

3.2 Adaptive Search Area

The GA search method described above works efficiently for offline images and online images with at most two faces present. Since the search region is fixed, the search takes about the same time in all frames. As the number of faces increase, there is need to further improve the search speed. One way is to adaptively control the search area. This is accomplished as follows.

1. The system is run on the first frame inside the face candidates.
2. Select the top six chromosomes and from their positional data, find the minimum and maximum x and y positions. Use this data to adjust the search area.
3. On the next frame, inherit the values from (2) and use them as the new face candidate region.

Although this method is simple, it produces a very adaptive search area that ensures that the GA converges even though the number of generations is only ten. Also note that, all the information contained in the top six chromosomes is inherited into the next frame and used in its initialization. This aids the convergence since we are starting with better chromosomes in the succeeding frames.

4 Results

4.1 Experiment Conditions

After the system was fully trained, real time simulations were carried out. For initial testing purposes, the simulations were carried out using printed faces images of five people pasted on the wall of the laboratory. Three of the faces are hung upright and the other two are pasted at minus 25^0 and 30^0 respectively. The face rotation is done so as to show that this system is invariant to frontal orientation of +/- 30 degrees as dictated by the genetic algorithm. Also note that an image similar to skin color is also included for contrast. The capture device is the BWC-130H01/SV USB camera from Bufallo Inc. The initial distance between the camera and the images is 2 meters. This experiment was carried out on a Dell Precision M60 laptop computer. The image size used in this work is 320x240pixels.

4.2 Simulation Method

The system testing followed the following steps:

1. After the image is captured, run the skin color detector to extract the skin color regions. These regions are the face candidates, Fig. 1. This continues for five consequtive frames until the face candidates are stable.

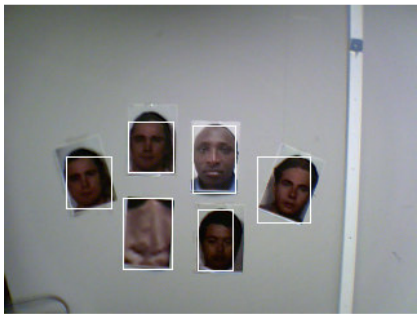

Fig. 1. Skin color region (Face candidates)

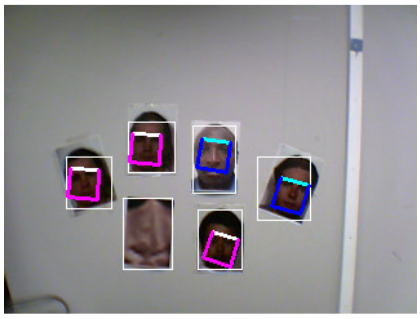

Fig. 2. Initial face locations detected inside the face candidates.

2. The genetic algorithm is then run inside each face candidate sequentially. The face samples extracted have their fitness individually calculated using the neural networks. The genetic algorithm processes of selection, reproduction and mutation follow. A sample result is shown by Fig. 2. This process continues for about three frames.
3. After step 2, the areas in which the best six face samples are extracted from becomes the new face candidates. After this until the last frame per second, Step 2 is the run only once inside the shrunken face candidates, Fig. 3.

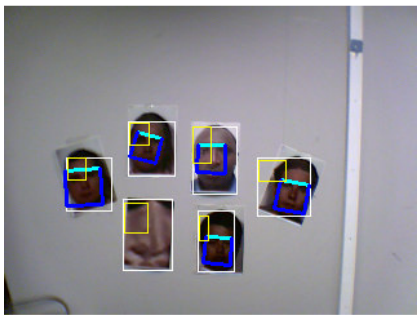

Fig. 3. Adaptive search area control. Notice that the original face candidate area has been reduced to less than a quarter of the original size.

4. Steps 1 to 3 are then repeated every second. Therefore, original face candidates extraction is done only once per second.

4.3 Final Results

Table 2 shows the average processing time taken for each step. (Note that the times shown below do not include the time taken to capture and draw the frames.) It is the time taken to run our system on the captured frame data. (The real time taken per frame is between 70 and 140 milliseconds).

Table 2 Face detection steps times

	Average Time Taken/ Frame (Msec)	System Average (Msec)
Step 1	15	27
Step 2	35	
Step 3	23	

The final system accuracy was calculated using the results from five minutes of testing. The number of frames processed was 8920 frames (Note that the total memory required to save these images is about 2GB and the total real time taken is about 20 minutes).

About 30 frames are processed per second, five of which are not searched for faces. Therefore, the total frames to be searched for faces are about 7300.

We carried out two sets of experiments, one with the camera at two meters (all five faces are visible), (System 1) and the other with the camera only one meter from the faces (only three faces within the camera view) (System 2), Fig. 4.

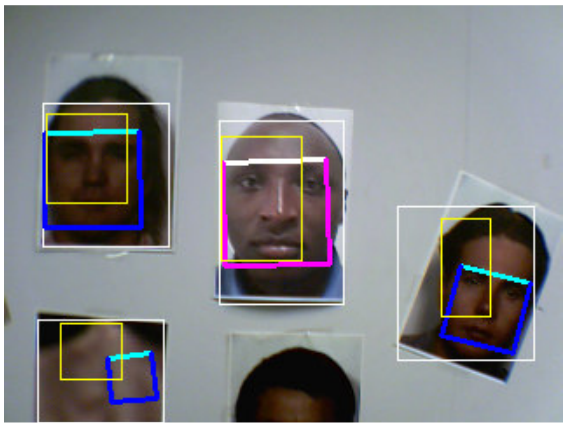

Fig. 4. Detection results with the camera 1 meter from the faces.

The systems accuracy, false acceptance rate (FAR) and false rejection rates (FRR) are as shown in Table 3.

Table 3 Final System Results

	FAR (%)	FRR (%)	Accuracy (%)
System 1	0.01	99.9	97.7
System 2	2.0	98	95.6

This system does not only extract the face from an image, it also provides information about the face's position, size and orientation. We think that this information could be useful to other applications. The face parameters, that is, position, size and orientation were all found to be within 5% error compared to the

real ones. Therefore, it can be said that the faces were accurately extracted.

4.3 Discussion

There is 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, then 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 large. Therefore, the search space is large too. Two, this system is trained with images 30x30pixels in size. In system 2, the face samples are almost twice this size and we think that this contributes to the lower detection accuracy.

5 Conclusion

In this paper, a real time face detection method using several small size neural networks and a genetic algorithm with adaptive search area control is proposed. From the experimental results, the final real time face detection average accuracy of 97% was achieved. Fastest real time operation of about 15 frames per second was realized. The system's ability to reject non-faces is about 1.9%. These results are based on five 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.

Future works include testing this system on real subjects and the improvement of the system to be more robust to the camera position. To reduce false acceptance rates, template matching could be used to reconfirm the final results.

References:

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

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

[3] 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*, 1997, pp. 130-136.

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

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

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

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

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