A Neural Network Face Recognition System

Aitkenhead, M. J.*¹, McDonald, A. J. S.²

¹The Macaulay Institute, Craigiebuckler, Aberdeen AB15 8QH, UK
²Department of Plant & Soil Science, University of Aberdeen, Aberdeen AB24 3UU, UK

*Corresponding Author – email m.aitkenhead@macaulay.ac.uk

Abstract

A neural network based facial recognition program (FADER – FAce DEtection and Recognition) was developed and tested. The hardware and software components were all standard commercial design, allowing the system to be built for minimal cost. Using a set of 1000 face and 1000 ‘no-face’ images, we achieved 94.7% detection rate, and a 0.6% false positive rate. Three different neural network models were applied to face recognition, using single images of each subject to train the system. A novel adaptation of the Hebbian connection strength adjustment model gave the best results, with 74.1% accuracy achieved. Each of the system’s components, including an intermediate substructure detection network, was subject to evolutionary computation in order to optimise the system performance.

Keywords
Neural networks, face recognition, substructure detection, evolutionary computation.
Introduction

There are a multitude of reasons why an observer may wish to detect or identify an individual person in a scene. O’Malley et al. (2002) discuss human tracking using video camera technology, for example in airports where there are areas that are off-limits to unauthorised personnel, and describe a method that relies on motion detection and clothing colour to first recognise that a person is within the field of view and then track their location. Trivedi et al. (2000), in demonstrating an ‘intelligent environment’, showed that human tracking and recognition will be important aspects of any future system that allows humans and computers to interact in more natural and efficient ways.

To this end, automated recognition of an individual from an image of their face has received a great amount of study in recent years, to the point where it has been mooted that the problem has been more or less solved. Gross et al. (2001) suggest just this, but go on to argue that there are still some areas in the topic that can be improved. While face recognition can be accurately, rapidly and consistently accomplished in controlled situations, we would argue that the technology is still not flexible and generic enough to deal with day-to-day, realistic situations.

Essa (1999) gives a good account of the motivations for computer-based facial recognition, and emphasises the difference between the problems of detecting faces and recognising them. This difference is also mentioned by Aas (1998), who used the difference between two frames (motion detection) to detect subjects. We will examine the use of motion detection to eliminate background clutter in a later section.

Fromherz et al. (1997) separates the problems of facial recognition into three categories: frontal, profile and view-tolerant. In this work we develop a view-tolerant
approach that requires both the mouth and both eyes to be visible in the image, so while
the system is capable of dealing with more than simple frontal images, it cannot deal with
every facial image. We also require that the system be capable of dealing with generic,
cluttered and novel scenes, because while it is easier to deal with a blank background this
situation will occur rarely in real situations. Weng (1998) gives a good discussion of the
challenges facing developers of robotic systems capable of interacting with their
environment with the same skill as humans.

Another problem is that in real situations, it will usually be desirable to compare
an unknown face against a single example of a identified image. Using multiple images
to train a system may not be possible, in only for the reason that only one image of a
subject exists. For example, police may want to identify someone in an image from
CCTV cameras, and might only have a single ‘mug-shot’ in their records. One research
area that differs slightly from face recognition is that of image retrieval, but there are
areas of overlapping interest between this topic and that of recognising a face from a
single image. If the problem of recognising an unknown person is redeclared as ‘find
someone who looks like this’, then the problem becomes one of image retrieval.

Di Sciascio et al. (2002) discuss the problems of retrieving images from high-
level characteristic descriptions (objects within an image), rather than from low-level
characteristics such as colour distribution or texture. With faces, low-level characteristics
may not be enough to allow correct recognition. Colour and lighting are variable, and
image quality and texture may also differ between images. For high-level characteristic
recognition, a hierarchical method of feature extraction that builds upwards from simple
structures to more complex ones is required. In this work, a substructure-detecting neural network design similar to that used by Aitkenhead & McDonald (in press) is used.

Kirchberg et al. (2002) used the Hausdorff method for detection of frontal face images. This method uses a predefined edge model of the human face to detect candidates in images, and was adjusted using genetic algorithm methods and existing facial images to modify a simple ellipse model. The use of a genetic algorithm is particularly useful in situations such as facial recognition, where an exact definition of the system being studied is not available.

One of the most common methods of facial recognition is that of Principal Component Analysis (PCA), which involves measurements of spatial relationships between important features of the face, for example eyes and mouth. Romano et al. (1996) used PCA to perform real-time face verification, allowing users access to a computer. Neural networks are a less common method used in facial recognition, no doubt partially due to the successes that have been shown by techniques that involve the use of PCA and eigensurfaces.

The major difference between neural network function and that of other methods is that NNs can be used in a hierarchical manner, detecting not only the most basic of patterns but also more patterns of patterns, and so on. Other techniques of object recognition commonly use different methods to detect different levels of structure within an image.

Increased computing power has allowed the development of neural networks capable of examining images in sufficient detail to recognise quite complex shapes. Talukder & Casasent (2001) demonstrated a neural network derived from eigenvectors.
capable of recognising facial poses and individual faces, using only one example of each original face to develop the eigenvector information, and thus proved beyond doubt that NNs are capable of facial recognition.

Here we aim to go one step further and train the neural network system itself with training images. Franco & Treves (2001) demonstrated a neural network based facial expression recognition system using the Yale Face Database (Belhumeur & Kriegman, 1997). Their method gave higher accuracy than PCA, and used a self-organising NN. The Yale Face Database uses images with a plain background and centred, frontal images, eliminating the need for face detection and localisation. These issues will also be addressed in this work, using a set of images that are not as ‘clean’ as the Yale Database.

Hjelmas & Wroldson (1997) and Huang & Wechsler (1999) applied PCA to reduce the dimensionality of images in face recognition work that also used neural networks. Using Principal Component Analysis in this way allowed the distinguishing features of the faces to be identified. While PCA does demonstrably allow substructures of images to be distinguished, it is felt that neural networks themselves also have this capability and can accomplish the pattern recognition process from the lowest level to the highest.

Evolutionary programming is a generic term used to cover a multitude of methods, in which programs are subject to mutation pressures and fitness selection. Kool (1999) gives a good literature review of both evolutionary programming and its use with neural networks. Here we aim to subject the neural networks used for facial detection and recognition, in order to optimise their performance.
Physical System

Image capture was performed using a Logitech Quickcam with USB connection, set to image resolution of 320x240. Screen capture rate was limited by the custom Visual Basic program written, with a capture rate of ~0.4 Hz being achieved on a 1.2 GHz PC. The camera was capable of dealing with a wide range of light levels, taking approximately 3 seconds to adjust to rapid illumination change. All programs were written using Microsoft Visual Basic 6.

The shareware video capture control ‘ezvidcap.ocx’ developed by Mercer (1998) and used for screen capture is available at http://www.shrinkwrapvb.com/ezvidcap.htm. Two versions are available, for Visual Basic versions 5 and 6. The control allows the user to perform several operations on the video stream, including single frame capture, and gives access to the Video For Windows functionality of Microsoft Windows 95 and 98. Full documentation of the software is also available at the website address given.

Figure 1 gives a screen shot of the FADER program, showing the incorporated video camera stream and the latest captured image.

In order to train the system, images were taken from a group of volunteers comprising twenty people. For each volunteer a set of five images was taken. Sex and age were varied throughout the volunteer group, while within the set of images taken of each volunteer, facial expression, facial position and orientation within the camera frame and camera position (background) were also not constant. In each case the location of the subject’s face within the image was limited only by the restriction that boundary of the image should not intersect with the boundary of the subject’s head. The orientation of the face was restricted so as to be within 45° of the direction to the camera in both rotation
and declination. All volunteers were seated and were between 1 and 2 metres from the camera when the images were taken.

The only physical parameter that was kept approximately constant throughout the acquisition of the training images was background lighting, although the light source varied from natural sunlight to that supplied by overhead strip lights and bulbs. The webcam software associated with the sampling program was set to automatically adjust contrast and brightness levels. Each training image was initially 160x120 pixels in size, with RGB in the range 0-255 (approx 16 million colours). No light sources were within the field of view of the camera when images were taken, although in several cases there was reflection caused by background objects. Figure 2 shows several examples of images used in training and identification.
Neural Networks

FADER was actually composed of three discrete networks, with the outputs of network 1 supplying the input for network 2, and the output for network 2 supplying the input for network 3. Network 1 was used to detect faces within the images, while network 2 detected substructures within the captured facial images, and network 3 performed the actual recognition. Figure 3 shows the modular structure of the system.

Face Detection

Neural network 1 was composed of an input layer of size 32x32, two hidden layers of variable size (with upper and lower limits of 100 and 1000 nodes, respectively) and an output layer consisting of 2 nodes, corresponding to presence or absence of a face. Each layer was fully connected to its successor, and the backpropagation training algorithm was used to adjust the connection weights.

Training of the face detection network was done by providing the system with face images adjusted using a method given by Rowley et al. (1996), in which the eyes and the center of the upper lip were located manually and used to adjust the image as follows:

1. Rotate the image so that both eyes are on a horizontal line.
2. Scale the image so that the distance from the point between the eyes to the centre of the upper lip is 20 pixels.
3. Extract a 32x32 pixel region, centered on the point equidistant between the middle of the line between the eyes and the upper lip.

(adapted from Rowley et al., 1996).
An original set of 200 faces was used to produce a total of 1000 face images by rotating each image clockwise and anticlockwise through 15° and 30°. The procedure of manually locating faces in images was accelerated through the use of a program that opened each of the images in turn and allowed the user to click on the three locations in question. A corresponding set of 1000 non-face 32x32 grids was created by capturing 20 images with the webcam in different orientations in a cluttered environment, and extracting 50 32x32 grids from each, selecting the top-left pixel location of each at random. The image arrays given to the neural network alternated between ‘face’ and ‘no face’ images. A simple edge detection algorithm was used to transform each training image, as explained in the System Application section. In Figure 4 the sequence of operations (not including edge detection) on a face detection training image is demonstrated, using an uncluttered image for demonstration purposes.

Training was carried out for a total of 10000 steps, with the network’s control variables (number of nodes in the two hidden layers, difference between positive and negative face presence output values) being subject to mutative evolutionary pressures as described in the following Control Variable Evolution section. Following the evolutionary process which lasted for 1000 generations, the face detection network was found to be capable of correctly detecting a face in one of the training images 94.7% of the time, with a false positive rate of 0.6%.

Substructure Detection

In recent years there has been strong discussion over whether facial recognition amongst humans depends more on information gleaned from individual spatial features or from the layout of the spatial configuration of these features. Cabeza & Kato (2000)
provide an argument for both sides, showing that both configural and ‘salient feature’
processing have a part to play in facial recognition. Walker et al. (1998) define salient
features as ‘those which have a low probability of being mis-classified with any other
feature’. In other words, they should be relatively simple and consistent in shape. The
problem remains with how the eye and the mind detect these constituent features of a face
or their spatial relationships.

Given a pixel-by-pixel description of a subject’s face, it is necessary to build up a
hierarchy of information, starting with the simplest features and working upwards. The
use of neural networks in substructure detection has been shown to work in the past. Liu
& Wang (2001) discuss the importance of feature extraction and perceptually meaningful
representations to classification problems, including facial recognition.

Everingham et al. (1998) used NNs to classify objects following feature
extraction at lower level by other methods. Here we use a method of substructure
detection that, it can be argued, supports both sides of the feature/configuration divide,
first detecting features and then passing their description up to a level that integrates
information about their structure and position into a neural-network-based description of
the subject’s face. Here we aim to detect salient features using a neural network from the
very start, and construct a hierarchical description of the image.

Many different kinds of neural network methodology exist in the literature. One
of the most commonly used is the backpropagation algorithm, which was found to work
extremely well in face detection. For substructure detection as applied to face
recognition however, the best method will be one that is capable of working around
missing or hidden information, for example in the situation where part of a person’s face
is masked by clothing. Auto-associative neural networks are arguably the best technique for this potential problem. In this general class of NN, connection strengths are adjusted according to the activation levels of connected nodes. Bohme et al. (1999) used AANNs to reconstruct missing sensor data in a water treatment plant.

The selected 32x32 grid from the image was translated onto an input layer of the same size. Training was carried out using an adapted reinforcement technique, which was based upon Hebbian connection modification in which the connection effectiveness is increased by simultaneous preconnection and postconnection activity. This adapted technique is similar to that described by Linsker (1986).

In addition to the input layer there were two hidden layers, of original size 250, with nodes assigned random x and y coordinates so that they lay within a cell one arbitrary unit on a side. The x-y plane distance between nodes in neighbouring layers was determined and used to evaluate their connectivity. Each layer was given two control variables used to determine connectivity – a maximum x-y plane separation $d_{\text{max}}$ between nodes in that layer and nodes in the next, and a connectivity value $c$ between 0 and 1. If a node in layer $n$ lay within the maximum separation distance from layer $m$ (where $m=n-1$), then the connectivity variable was used to determine at random whether the two nodes were connected. This resulted in a columnar structure within the neural network architecture, comparable to that described in Linsker (1988a) and Linsker (1988). For each of the layers the maximum separation variables were initially set at 0.2, and the connectivity variable at 0.5. Figure 5 depicts the columnar structure that results from this architecture.
Nodes in each layer in turn were activated by the inputs from nodes in the previous layer. The activation signal $A_j$ for each node $n_j$ from all $m$ nodes connected to it is calculated using equation 1:

$$A_j = \sum_{i=m} a_i s_{ij}$$  \hspace{1cm} (eqn. 1)

where $a_i$ is the activation level of node $i$ (0 or 1), and $s_{ij}$ is the connection weighting from node $i$ to node $j$. The connection weighting falls within a range $s_{\text{min}}$ to $s_{\text{max}}$, with each layer having initial values for these variables of $-1$ and $1$, respectively. Node activation resulted if the input ($A$) was over the threshold in each case, and the individual node thresholds were adjusted up or down accordingly, according to the relationship given in equations 2a and 2b. The maximum possible threshold, $t_{\text{max}}$, was initially set to 2, and $t_{\text{min}}$ was set to 1.

$$t_1 = t_0 + \alpha(t_{\text{max}}-t_0) \quad [A>t_0]$$ \hspace{1cm} (eqn. 2a)

$$t_1 = t_0 + \alpha(t_{\text{min}}-t_0) \quad [A<=t_0]$$ \hspace{1cm} (eqn. 2b)

Here, $\alpha$ is an arbitrary constant whose value was initially set to be 0.01, giving relatively rapid learning without sacrificing accuracy.

Following activation of the substructure network’s first layer and propagation of the activation signal through the network, connection and node modification occurred. For nodes $i$ and $j$, with connecting connection weight $s_{ij}$, the following set of equations is applied:

$$s_{ij} = s_{ij} + \alpha(s_{\text{max}}-s_{ij}) \quad [a_i = 1, a_j = 1, s_{ij} > 0]$$ \hspace{1cm} (eqn. 3a)

$$s_{ij} = s_{ij} + \alpha\beta(s_{\text{max}}-s_{ij}) \quad [a_i = 1, a_j = 1, s_{ij} < 0]$$ \hspace{1cm} (eqn. 3b)

$$s_{ij} = s_{ij} + \alpha\beta(s_{\text{min}}-s_{ij}) \quad [a_i = 1, a_j = 0, s_{ij} > 0]$$ \hspace{1cm} (eqn. 3c)

$$s_{ij} = s_{ij} + \alpha(s_{\text{min}}-s_{ij}) \quad [a_i = 1, a_j = 0, s_{ij} < 0]$$ \hspace{1cm} (eqn. 3d)
These equations can be progressively explained using a set of rules:

1. The connection $s_{ij}$ will be adjusted asymptotically towards either $s_{\text{max}}$ or $s_{\text{min}}$, depending upon the value of the receiving node, $a_j$. For adjustment towards the value of $s_{\text{max}}$, this means an increase proportional to $(s_{\text{max}} - s_{ij})$, while for adjustment towards $s_{\text{min}}$, this means an increase proportional to $(s_{\text{min}} - s_{ij})$, which will be negative.

2. The factor $\beta$ (an initial value of 0.1 was set after trial and error) is used where the values of the connection $s_{ij}$ and the receiving node activation $a_j$ do not correspond; the connection may be negative (inhibitory) while the node it sends stimuli to is still receiving enough of a stimulus from other connections to be activated. Introducing the factor $\beta$ reflects the fact that a particular connection is not always responsible for a node’s activation or lack of activation, and in these cases this connection’s adjustment should be smaller than for the connections that are responsible for the node activation.

**Facial Recognition**

Three different methods were applied to the recognition of faces from the output of the substructure neural network. As the output would take the form of a set of nodes either active (1) or inactive (0), the original intention was to use a Hopfield neural network to develop attractors that would correspond to known faces. The output layer from the substructure detection network was used as the input for the facial recognition attractor, a Hopfield neural network using Hebbian connection modification. This network was fully connected and consisted of between 100 and 500 nodes (arbitrary upper and lower limits), with a connection weight modification rate of 0.05. For connection from the substructure network, each node in the network was given a random
location in the x-y plane, with minimum and maximum values of 0 and 1 respectively. During training, for each input set the network was simultaneously updated only once, while in testing it was allowed to oscillate through 10 steps before the state was examined.

The final step of the training session involved showing the system the training example of each subject and allowing the recognition network 10 steps to settle, before examining the set of active and inactive nodes. In identifying an unknown face the recognition network’s node activity set was compared to the set corresponding to each training image, and the Hamming distance between the two sets measured. The training image that gave the lowest Hamming distance to the unknown image was then declared to be the match.

It was suspected that due to overlap of the sets of active nodes, the effectiveness of the classical Hopfield NN model would be reduced. The second method used in facial recognition involved simply having a layer of unconnected nodes, without the Hopfield NN, and comparing the list of active and inactive nodes generated by the ‘known’ face against that generated by the ‘unknown’, measuring the Hamming distance between each and taking the closest match as the recognised image. This simpler method also allowed for determination of the efficacy of the Hopfield network in distinguishing between overlapping attractors.

The third method that used the substructure network output to identify faces was an extension of that network, as a single self-connected layer that obeyed the same node and connection adjustment rules but which by being self-connected was used to form attractors and thus recognise subjects. This method allows comparison with the simpler
Hopfield/Hebb attractor network in terms of attractor formation and recall, and was also subjected to the evolutionary processes that formed the substructure detection network.
Control Variable Evolution

In order to optimize the neural network’s performance, adjustments of the variables involved were necessary. It immediately became obvious that with the large number of control variables present (3 for the face detection network, 23 for the substructure network, between 1 and 7 for the facial recognition network depending upon type) the scope for variation was too large to be confident of reaching optimal values after a trial-and-error search through the multidimensional variable space.

A system was required that could deal with a different numbers of volunteers. Rather than evolving a different set of control variables for each of the different numbers of volunteers used, each evolutionary step was carried out with a random number between two and twenty. Using this method it is possible that a system was not achieved that would give the best results for one particular number of subjects. However, it would not be difficult to take the control variable values for the generalised system and evolve them for a situation that used a specific number of subjects. This was not done here as it was considered extraneous to the work being carried out.

The method used to evolve the neural networks used in this work was kept relatively simple. Using the selected ‘parent’ set of control variables, we created five ‘children’, each with slightly different values from that of the parent. The children were then trained individually and a measure of fitness taken. The child with the highest fitness score was then used as the parent for the next generation. The mutation in the control variables between the parent and each child was a random increase or decrease by 1% of the predefined maximum and minimum values of these control variables. Table 1 gives the final values arrived at for the facial detection network, while Tables 2, 3 and 4
give the values for the substructure/recognition networks for each of the recognition
network methods used.

We evolved the facial detection system first in isolation, then kept its control
variables fixed while we evolved the final two networks. Because three different facial
recognition network types were used, we carried out this secondary phase three times,
allowing the substructure network to evolve differently in each case. For the facial
detection network fitness was defined as the value of (correct positive detections – false
positive detections). For the substructure and recognition networks the fitness was
defined as the correct score, given as a percentage, of faces identified correctly.
System Application

The first stage in the facial detection training procedure for each image was the transformation from a colour image into greyscale (pixel range 0-255). Following this, the image was transformed into a 32x32 image and edge detection carried out on each pixel using a simple method of taking the standard deviation of the 3x3 group centred on the pixel in question. The value obtained for the standard deviation was divided by the maximum standard deviation for the 3x3 group to lie on the range [0,1]. Manjunath et al. (1996) describe this first stage of feature detection as being similar to the behaviour of the hypercomplex cells in the visual cortex. These cells are sensitive to line segments, endings and general changes in curvature.

The edge-detection image was then sampled at multiple points using square grids of a range of sizes from 16 pixels across to 128 across, using progressing doublings in size. Distances between the top-left point of each sample equalled one-eighth of the sample grid size, giving a total of 22021 samples from each 320x240 image. Each sample was adjusted in size to a 32x32 grid and given as an input to the face-detection neural network, which determined whether or not a face was present in the sample. If the positive ‘face present’ response from the neural network minus the negative ‘face absent’ response was over the evolved threshold value then the location of the image sample was translated into corner coordinates on the original 320x240 image. If an image sample was selected that has more than 50% overlap with an earlier selected sample then it was rejected.

Once the boundaries of the subject’s face had been established within the image, This square set of pixels was sampled and rescaled to a size of 32x32, and then
transformed using the same edge detection method given above. This final image was then used as the input for the substructure detection neural network, which fed forward to the facial recognition layer.
Results

Face detection

Following the evolutionary process, a backpropagation network of topology 1024:453:118:2 was found to give the highest success in face detection. The highest accuracy achieved using the 1000 faces obtained from 126 images was 94.7%, with 6 false positives obtained from the set of 1000 randomly generated images. Figure 6 shows a test run of the face detection network on an image obtained from the internet that had not been used to train the system. The image has been adjusted to fit the standard 320x240 size generated by the webcam. Of 16 faces in the image, the system located 13. The three faces that were missed by the system were all profile or nearly so. No false positives were located.

A timed run using only the trained face detection network on a PC with a 1.2 GHz Athlon processor was found to take 924 seconds. Dropping the 16-pixel grid searches reduced this time to 157 seconds and is equally successful provided the subject is within 5 metres of the camera. It was found that for images taken of people between 0.5 and 1 metres from the camera, the face occupied a large enough proportion of the image to drop all but the 128-pixel scale search. This enabled the entire program to run, with no loss in performance, in a period of 2.4 seconds.

Face recognition – Hopfield network

Figure 7 shows how variation of subject count affected accuracy after completed evolution using a Hopfield attractor network as the final network. Ten runs were performed using each subject count. As was expected, the evolutionary process led towards a larger network, with highest accuracy being achieved when the node count was
hovering near 500. Increasing the node count to 1000 did not increase the mean accuracy of the system by more than 1.3% for any subject count.

**Face recognition – Hamming distance**

In Figure 8, accuracy versus subject count using the basic Hamming distance measure for face recognition is shown. Here, the system evolved a final layer of size 265, with performance being generally unaffected when layers of larger node count were used.

**Face recognition – Substructure network**

Using the extended substructure detection network to develop face attractors gave uniformly better results than the standard Hopfield/Hebb neural network model. Table 2 gives values for the control variables of this third network following 1000 steps of evolution. The comparison of accuracy versus subject count is shown in Figure 9.
Discussion

While Hopfield/Hebb networks are capable of limited attractor recall with overlapping sets, and are better than simple Hamming distance comparison, a novel method that uses more complex node/connection interactions has been shown to improve on this ability. A combination of rapid node and slow connection adjustment gives a smaller network improved attractor formation abilities than a larger. Consideration of connection sign and node activation in connection weight adjustment also improves the capabilities of the network.

In recent years research into facial recognition has mostly been carried out using methods that rely on measurements of facial parameters, rather than neural network techniques. These methods have been proven to work well but are often limited in their capabilities. A unified method that is capable of dealing with single training images, extraction of faces in a cluttered and variable environment and identification using a large database of images with rapid processing times would prove extremely useful.

Here we have demonstrated a technique that accomplishes all of these goals bar one. As yet the system cannot be said to operate in ‘real-time’, with an image processing time of several minutes. The vast majority of this time is taken up by the face detection procedure looking for ‘small’ faces (far away from the camera), and can be eliminated if subjects are assumed to be within a certain distance. This will be a safe assumption for many situations that involve security checks, but will not be valid for others, e.g. crowd monitoring.

Determination of the location and size of a human face against a cluttered background, in order for the enlargement and centering of that face in the image to be
achieved, has turned out to be a relatively simple problem to overcome. Feature extraction is also made possible through neural networks. The remaining problem, actual recognition from a large database, is more difficult and further work needs to be done in this area to improve the system’s performance.

It is accepted that the level of accuracy given by the FADER system in its present form is not good enough for it to be considered for use in a practical application. However, this is an example of facial recognition using a system inspired by biological learning, rather than mathematical techniques, and uses a neural network much simpler than the human brain. It is felt that increased visual resolution and network size would improve performance. Brause (1995) gives a good argument for the use of this kind of neural network learning in favour of the backpropagation method, stating that since we do not properly understand the internal behaviour of biological learning systems, we cannot rely on any ‘special’ feedback patterns that do not occur in nature.

One objection to using neural networks for this kind of work is that they are ‘inscrutable’ – that is, it is difficult to understand why they make the decisions that they do in particular situations. This ‘black box’ analysis often proves problematic where a symbolic, logical explanation of the decision-making process is required. Priss (2001) discusses the difference between ‘formal concepts’ and the ‘associative concepts’ of fuzzy logic or neural networks, and argues that in any decision-making process a method that uses both types of concept would be better than one that restricts itself to either.

We agree with this viewpoint and argue that the neural network approach to face recognition should be augmented by some symbolic, logical method that allows for an explanation, if required, of what factors influenced the system’s decision. Zhou et al.
(2002) describe a method of extracting rules from neural network ensembles, thus allowing the reasoning process of the networks to be examined in detail. This approach might well prove useful in the future for determining which features are more useful in face recognition than others.
References


2nd European Conference on Disability, Virtual Reality and Associated Technologies, 183-192.


Technical Biographies

Matthew Aitkenhead is currently working on a Ph.D. on the use of Artificial Intelligence in modeling complex systems, at the Department of Plant & Soil Science, University of Aberdeen, Scotland. He has interests in autonomous agent design and environmental system modeling.

Dr A. James S. McDonald graduated B.Sc. Ecological Sciences and Ph.D. Electrical Engineering. He has a track record in plant physiological research in relation to plant nutrition and water relations. He is currently exploring the use of mutable Individual Based Models (IBM) in studying the dynamics of plant communities.
**Figure captions**

Figure 1. Screen shot of FADER system.

Figure 2. Subject image samples taken with webcam.

Figure 3. Modular structure of FADER system.


Figure 5. Representation of columnar structure of substructure detection neural network.

Figure 6. Test example of the facial detection system. A – original image, B – edge detection, C – face detection.

Figure 7. Comparison of subject count and system performance using Hopfield network.

Figure 8. Comparison of subject count and system performance using basic Hamming distance measurement.

Figure 9. Comparison of subject count and system performance using extended substructure detection network.
Tables

Table 1. Evolved control variables for the face detection network

<table>
<thead>
<tr>
<th>Control variable</th>
<th>Evolved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2 nodes</td>
<td>453</td>
</tr>
<tr>
<td>Layer 1 nodes</td>
<td>118</td>
</tr>
<tr>
<td>Detection threshold</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Table 2. Evolved control variables for the substructure/recognition networks using the Hopfield net for face recognition

<table>
<thead>
<tr>
<th>Control variable</th>
<th>Evolved value</th>
<th>Control variable</th>
<th>Evolved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1 nodes (fixed)</td>
<td>1024</td>
<td>Layer 2 t&lt;sub&gt;max&lt;/sub&gt;</td>
<td>2.80</td>
</tr>
<tr>
<td>Layer 1 d&lt;sub&gt;max&lt;/sub&gt;</td>
<td>0.17</td>
<td>Layer 3 nodes</td>
<td>680</td>
</tr>
<tr>
<td>Layer 1 c</td>
<td>0.29</td>
<td>Layer 3 d&lt;sub&gt;max&lt;/sub&gt;</td>
<td>0.21</td>
</tr>
<tr>
<td>Layer 1 s&lt;sub&gt;min&lt;/sub&gt;</td>
<td>-2.4</td>
<td>Layer 3 c</td>
<td>0.13</td>
</tr>
<tr>
<td>Layer 1 s&lt;sub&gt;max&lt;/sub&gt;</td>
<td>1.6</td>
<td>Layer 3 s&lt;sub&gt;min&lt;/sub&gt;</td>
<td>-1.9</td>
</tr>
<tr>
<td>Layer 1 t&lt;sub&gt;min&lt;/sub&gt; (fixed)</td>
<td>0.5</td>
<td>Layer 3 s&lt;sub&gt;max&lt;/sub&gt;</td>
<td>1.8</td>
</tr>
<tr>
<td>Layer 1 t&lt;sub&gt;max&lt;/sub&gt; (fixed)</td>
<td>0.5</td>
<td>Layer 3 t&lt;sub&gt;min&lt;/sub&gt;</td>
<td>0.85</td>
</tr>
<tr>
<td>Layer 2 nodes</td>
<td>855</td>
<td>Layer 3 t&lt;sub&gt;max&lt;/sub&gt;</td>
<td>1.25</td>
</tr>
<tr>
<td>Layer 2 d&lt;sub&gt;max&lt;/sub&gt;</td>
<td>0.26</td>
<td>α</td>
<td>0.015</td>
</tr>
<tr>
<td>Layer 2 c</td>
<td>0.38</td>
<td>β</td>
<td>0.155</td>
</tr>
<tr>
<td>Layer 2 s&lt;sub&gt;min&lt;/sub&gt;</td>
<td>-2.8</td>
<td>Recognition layer nodes</td>
<td>500</td>
</tr>
<tr>
<td>Layer 2 s&lt;sub&gt;max&lt;/sub&gt;</td>
<td>1.7</td>
<td>Recognition layer α</td>
<td>0.064</td>
</tr>
<tr>
<td>Layer 2 t&lt;sub&gt;min&lt;/sub&gt;</td>
<td>1.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3. Evolved control variables for the substructure/recognition networks using the Hamming distance measurement for face recognition

<table>
<thead>
<tr>
<th>Control variable</th>
<th>Evolved value</th>
<th>Control variable</th>
<th>Evolved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1 nodes (fixed)</td>
<td>1024</td>
<td>Layer 2 $t_{\text{min}}$</td>
<td>1.60</td>
</tr>
<tr>
<td>Layer 1 $d_{\text{max}}$</td>
<td>0.22</td>
<td>Layer 2 $t_{\text{max}}$</td>
<td>2.15</td>
</tr>
<tr>
<td>Layer 1 $c$</td>
<td>0.32</td>
<td>Layer 3 nodes</td>
<td>540</td>
</tr>
<tr>
<td>Layer 1 $s_{\text{min}}$</td>
<td>-2.2</td>
<td>Layer 3 $d_{\text{max}}$</td>
<td>0.29</td>
</tr>
<tr>
<td>Layer 1 $s_{\text{max}}$</td>
<td>1.8</td>
<td>Layer 3 $c$</td>
<td>0.18</td>
</tr>
<tr>
<td>Layer 1 $t_{\text{min}}$ (fixed)</td>
<td>0.5</td>
<td>Layer 3 $s_{\text{min}}$</td>
<td>-1.4</td>
</tr>
<tr>
<td>Layer 1 $t_{\text{max}}$ (fixed)</td>
<td>0.5</td>
<td>Layer 3 $s_{\text{max}}$</td>
<td>1.2</td>
</tr>
<tr>
<td>Layer 2 nodes</td>
<td>735</td>
<td>Layer 3 $t_{\text{min}}$</td>
<td>1.50</td>
</tr>
<tr>
<td>Layer 2 $d_{\text{max}}$</td>
<td>0.24</td>
<td>Layer 3 $t_{\text{max}}$</td>
<td>2.15</td>
</tr>
<tr>
<td>Layer 2 $c$</td>
<td>0.32</td>
<td>$\alpha$</td>
<td>0.009</td>
</tr>
<tr>
<td>Layer 2 $s_{\text{min}}$</td>
<td>-2.0</td>
<td>$\beta$</td>
<td>0.130</td>
</tr>
<tr>
<td>Layer 2 $s_{\text{max}}$</td>
<td>1.9</td>
<td>Recognition layer nodes</td>
<td>265</td>
</tr>
</tbody>
</table>
Table 4. Evolved control variables for the substructure/recognition networks using the extended substructure network for face recognition

<table>
<thead>
<tr>
<th>Control variable</th>
<th>Evolved value</th>
<th>Control variable</th>
<th>Evolved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1 nodes (fixed)</td>
<td>1024</td>
<td>Layer 3 (d_{\text{max}})</td>
<td>0.24</td>
</tr>
<tr>
<td>Layer 1 (d_{\text{max}})</td>
<td>0.18</td>
<td>Layer 3 (c)</td>
<td>0.18</td>
</tr>
<tr>
<td>Layer 1 (c)</td>
<td>0.32</td>
<td>Layer 3 (s_{\text{min}})</td>
<td>-2.6</td>
</tr>
<tr>
<td>Layer 1 (s_{\text{min}})</td>
<td>-2.2</td>
<td>Layer 3 (s_{\text{max}})</td>
<td>1.6</td>
</tr>
<tr>
<td>Layer 1 (s_{\text{max}})</td>
<td>1.5</td>
<td>Layer 3 (t_{\text{min}})</td>
<td>0.75</td>
</tr>
<tr>
<td>Layer 1 (t_{\text{min}}) (fixed)</td>
<td>0.5</td>
<td>Layer 3 (t_{\text{max}})</td>
<td>1.4</td>
</tr>
<tr>
<td>Layer 1 (t_{\text{max}}) (fixed)</td>
<td>0.5</td>
<td>Layer 4 nodes</td>
<td>445</td>
</tr>
<tr>
<td>Layer 2 nodes</td>
<td>820</td>
<td>Layer 4 (d_{\text{max}})</td>
<td>0.38</td>
</tr>
<tr>
<td>Layer 2 (d_{\text{max}})</td>
<td>0.23</td>
<td>Layer 4 (c)</td>
<td>0.27</td>
</tr>
<tr>
<td>Layer 2 (c)</td>
<td>0.32</td>
<td>Layer 4 (s_{\text{min}})</td>
<td>-3.65</td>
</tr>
<tr>
<td>Layer 2 (s_{\text{min}})</td>
<td>-2.2</td>
<td>Layer 4 (s_{\text{max}})</td>
<td>2.25</td>
</tr>
<tr>
<td>Layer 2 (s_{\text{max}})</td>
<td>1.55</td>
<td>Layer 4 (t_{\text{min}})</td>
<td>1.25</td>
</tr>
<tr>
<td>Layer 2 (t_{\text{min}})</td>
<td>1.55</td>
<td>Layer 4 (t_{\text{max}})</td>
<td>1.9</td>
</tr>
<tr>
<td>Layer 2 (t_{\text{max}})</td>
<td>1.70</td>
<td>(\alpha)</td>
<td>0.018</td>
</tr>
<tr>
<td>Layer 3 nodes</td>
<td>735</td>
<td>(\beta)</td>
<td>0.145</td>
</tr>
</tbody>
</table>
Figure 1
Figure 3
Figure 4
Figure 7
Figure 8
Figure 9