Machine Learning Research
2017; 2(4): 152-168
http://www.sciencepublishinggroup.com/j/mlr
doi: 10.11648/j.mlr.20170204.16

Radial Basis Function Neuroscaling Algorithms for Efficient Facial Image Recognition

Vincent A. Akpan[1, *], Joshua B. Agbogun[2], Michael T. Babalola[3], Bamidele A. Oluwade[4]

[1]Department of Biomedical Technology, The Federal University of Technology, Akure, Nigeria

[2]Department of Computer Science, Kogi State University, Anyigba, Nigeria

[3]Department of Physics Electronics, Afe Babalola University, Ado-Ekiti, Nigeria

[4]Department of Computer Science, University of Ilorin, Ilorin, Nigeria

Email address:

vaakpan@futa.edu.ng (V. A. Akpan), ajbjoshua@gmail.com (J. B. Agbogun), mtbablola@abuad.edu.ng (M. T. Babalola), deleoluwade@yahoo.com (B. A. Oluwade)

[*]Corresponding author

To cite this article:

Vincent A. Akpan, Joshua B. Agbogun, Michael T. Babalola, Bamidele A. Oluwade. Radial Basis Function Neuroscaling Algorithms for Efficient Facial Image Recognition. *Machine Learning Research*. Vol. 2, No. 4, 2017, pp. 152-168. doi: 10.11648/j.mlr.20170204.16

Received: September 30, 2017; **Accepted**: November 9, 2017; **Published**: December 28, 2017

Abstract: A Radial basis function neural network-based probabilistic principal component analysis (RBFNN-PPCA) on image recognition based on facial recognition was made. The variational properties of face images are investigated with Eigenfaces algorithm to validate the proposed RBFNN-PPCA algorithm and technique for enhanced optimal image recognition system design. Ten different face image samples for each one hundred different individuals with their corresponding bio-data were taken under different light intensities were cropped and pre-processed. The resulting one thousand face image samples were split into 80% as the training set which constitutes the database of known face images and 20% as the test set which constitutes unknown faces images. Analysis was made on the one thousand face images based on the proposed RBFNN-PPCA algorithm and the Eigenfaces algorithm. The two algorithms were applied simultaneously for enhanced optimal face recognition, and the simulation results show that the proposed face image evaluation techniques as well as the proposed RBF neuroscaling algorithm recognizes a known face image or rejects an unknown face based on the database contents to a high degree of accuracy. The proposed face recognition strategy can be adapted for the design of on-line real-time embedded face recognition systems for public, private, business, commercial or industrial applications.

Keywords: Eigenfacess, Face Recognition, Neural Network (NN), Neuroscaling, Probabilistic Principal Component Analysis (PPCA), Radial Basis Function (RBF), Singular Value Decomposition (SVD)

1. Introduction

Our faces are complex objects with features that can vary over time. However, we humans have a natural ability to recognize faces and identify persons in a glance. Of course, our natural recognition ability extends beyond face recognition, where we are equally able to quickly recognize patterns, sounds or smells. Unfortunately, this natural ability does not exist in machines, thus the need to simulate recognition artificially in our attempts to create intelligent autonomous machines. Intelligent systems are being increasingly developed aiming to simulate our perception of various inputs (patterns) such as images, sounds etc. Biometrics, in general, and facial recognition in particular are examples of popular applications for artificial intelligent systems. Face recognition by machines can be invaluable and has various important applications in real life, such as, electronic and physical access control, national defense and international security [1]. Simulating our face recognition natural ability in machines is a difficult task, but not impossible [2]. Throughout our life time, many faces are seen and stored naturally in our memories forming a kind of database. Machine recognition of faces requires also a database which is usually built using facial images, where sometimes different face images of a one person are included to account for variations in facial features. The development

of an intelligent face recognition system requires providing sufficient information and meaningful data during machine learning of a face.

Biometric identification is a general term for technologies that permit matches between a live digital image of a part of the body and a previously recorded image of the same part, usually indexed to personal or financial information [3]. Biometric identifiers include digital fingerprints [4], retinal scans [5], hand geometry [6], facial characteristics [7] and vocal patterns [8]. Biometric scanning systems typically do not record the entire imprint of a physical feature but only that portion, or template that should be time-invariant within some statistical limit. Since the body changes over time, the statistical algorithm must be elastic enough to match a stored image with a later live scan from the same person, without normally matching two similar individuals. This creates limitations on the uniqueness of the images, which are overcome by using multiple images from one person, or a biometric image plus other information. In some applications identities can be verified within a population of millions.

Twenty years ago the problem of face recognition was considered among the hardest in Artificial Intelligence (AI) and computer vision [9]. Surprisingly, however, over the last decade there have been a series of successes that have made the general person identification enterprise appear not only technically feasible but also economically practical [10]. The apparent tractability of face recognition problem combined with the dream of smart environments has produced a huge surge of interest from both funding agencies and from researchers themselves. It has also spawned several thriving commercial enterprises. There are now several companies that sell commercial face recognition software that is capable of high-accuracy recognition with databases of over 1,000 people [11]. These early successes came from the combination of well-established pattern recognition techniques with a fairly sophisticated understanding of the image generation process. In addition, researchers realized that they could capitalize on regularities that are peculiar to people, for instance, that human skin colors lie on a one-dimensional manifold (with color variation primarily due to melanin concentration), and that human facial geometry is limited and essentially 2-D when people are looking toward the camera. Today, researchers are working on relaxing some of the constraints of existing face recognition algorithms to achieve robustness under changes in lighting, aging, rotation-in-depth, expression and appearance (beard, glasses, makeup); which are problems that have partial solution at the moment [12].

2. Background Knowledge and Review of Related Work

2.1. Historical Background

A growing number of biometric technologies have been proposed over the past several years, but only in the past 5 years have the leading ones become more widely deployed. Some technologies are better suited to specific applications than others, and some are more acceptable to users. The concept of biometrics probably began with the human use of facial features to identify other people. Modern biometrics, however, started in the 1880s when Alphonse Bertillon, chief of the criminal identification division of the police department in Paris, France, developed a method of identification based on a number of bodily measurements, this system was then called The Bertillon System [13].

2.1.1. The Importance of Facial Biometrics

The growing need for enhanced security systems in government, commercial, and personal applications is increasingly being met by biometric identification methods. Biometric systems currently use fingerprints, iris, retina, palm vein, and faces for authentication. The two main functions of a biometric system are identification and verification. For several reasons, "facial recognition technology should be considered as a serious alternative in the development of biometrical or multi-biometrical systems, it requires no interaction from the user and no advanced hardware. In contrast to fingerprints and eyes, the face does not have as many unique features that can be compared. Areas that are of particular interest of the face and used for comparison are the "upper outlines of the eye sockets, the areas surrounding the cheekbones, the sides of the mouth, and the location of the nose and eyes. These areas are considered to hold the most distinct qualities of an individual's face. However, false acceptances are to be expected and are influenced by several factors such as lighting, an increase or decrease in weight, and aging [14].

2.1.2. Facial Recognition by Grid

Facial biometrics has been tested in different settings in the attempt to comprehend how different variables, when introduced, affect the outcome or validity of the testing. The purpose of using facial biometrics in a security situation may be to prevent imposters from attempting to break into a certain location. It is safe to assume that these people will be motivated to interfere with being correctly identified or attempt to alter their features in an attempt to gain access. In 1996, a computer science professor named Harry Wechsler developed technology to make facial recognition software more accurate based on individual components of the face [15]. This approach was known as the recognition by parts approach. This approach used individual face components along with sequential recognition. The program will take portions that appear to be most relevant and compare them. In this manner the program is able to distinguish one face from another through comparison and elimination [16].

2.1.3. Facial Biometric in Use

The interest and use of facial biometrics are widespread and has already been used in various arenas. After the terrorist attacks of September 11, 2001, there were increased talks about the use of this technology especially in airports. However, out of the 19 terrorists, only 2 were actually known to the Central Intelligence Agency (CIA) and Federal Bureau of Intelligence (FBI) leaving the database useless for the

unknown 17 identities [17]. Another point to take into consideration are that airports are full of fast moving crowds of people and a person would need to be still, unobstructed and close enough to the sensing camera to obtain an accurate photo. These are just a few reasons why airport use may not be as effective. A point worth mentioning is the potential for drivers' licenses or passports to be used. This would make up the best database to compare photos to. However, since it would be illegal to sell the photos to private companies, the government would need to find the most effective way to utilize this ready-made database. Facial biometric systems seem to be the wave of the future for both identification and authentication. Wide ranges of applications are on the horizon. Facial biometric systems will enable access to secure and sensitive areas, such as energy supply facilities, nuclear power stations or emergency service control centers. Digital e-cards are opening up new opportunities for facial controls in the areas of banking and business. Public demand for these applications may be the driving force behind further progress in biometrics research [14].

2.1.4. Aging and Facial Recognition

Aging is likely to be the most challenging of problems for biometric technologies. Inherent in facial biometrics is the human aging factor and the possibility of false identification or failure of identification, due to facial morphology. The degree of variation that occurs in the face during aging of adults certainly affects outcomes for facial biometric systems. These changes have been studied sporadically in anthropological research and have received less attention in biometric-related literature. According to [18], face recognition techniques perform best using images from well-controlled environments, but perform poorly when illumination varies in non-controlled environments. It is also challenging when large pose variations exist in the images due to their two-dimensional nature. To address this issue it is better to use three-dimensional images. Roadblocks to pervasive implementation of current face recognition technology are reasons for high error rates. A reason is attributed to 2-D technology that is the basis of most of the current applications on the market. 2-D technology measures height, width and distance between feature points to make an identification which is flawed since faces are 3-D, with irregularly shaped features -noses, lips, ears, hair - that change in appearance as the face turns. Faces also reflect light and produce shadows, essentially creating new and different images. With 2-D technology, failure rates rise with changes in facial posture or expression or variable lighting [19].

2.1.5. Advantages of 3-D Facial Recognition

Unlike its 2-D counterpart, 3-D face recognition uses the geometry of a subject's facial structure. 3-D range cameras are used to capture the depth of an object instead of the color. In this way, the potential exists to achieve more accurate results. Range cameras may use optical imaging technologies such as triangulation, interferometry, and imaging radar. Dr. Ramalingam [20] developed a new 3-D system that is able to capture detailed images of faces while in motion. This is viewed as a breakthrough because with previous systems, the subject needed to be within a controlled environment as well as stationary. Dr. Ramalingam stated that this system "applies new mathematical algorithms and a stereo camera setup". A stereo camera setup uses two cameras allowing two lines of sight to cross each other at a single focus point, similar to how humans view objects. This new system photographs sections of a subject's face and matches it to a template in the system. Ramalingam's 3-D system is also believed to be faster than other systems like it and also allows real-time capture [14].

2.1.6. Scene Geometry in Perspective Projection

Instead of assuming orthographic projection method, there is another class of single view reconstruction methods that use perspective projection cues to recover the 3D structures. These methods rely on assumptions in scene geometry. Horry and co-workers [21] reconstructed piecewise planar models based on user-specified vanishing points and geometric invariants. For example, they compute the perspective distortion of the parallel lines and other easy cues, in order to do the single view reconstruction. Their results are very interesting, for example, showing a virtual museum where you can walk into the 2-D paintings such that the structures have 3-D appearance [22] added automatic segmentation to separate objects in the scene. However, these methods require that certain geometric easy cues exist, such as lines, planar surfaces, squares and so on. It is not clear, however, how such methods could be generalized to general objects of free-form, such as faces [22].

2.1.7. Lambertian Assumption and Texture Mapping

In 1998, Vetter introduced two variations to the pure linear class method [23]. Firstly, the linear class approach was applied to the parts of a face separately. Secondly, a 3-D laser scanner to record 3D data of human heads was used, and averaged them to get a single 3-D model, which was used to establish pixel-wise correspondence between the two reference face images in the two different poses. This correspondence field allows texture mapping across the view point change, which leads to better quality of synthesized texture in the visible pixels, while the occluded pixels are still synthesized using linear class method.

Texture mapping techniques have also been used to synthesize novel views of faces [24]. He built the 3D model for each input image by finding the symmetric landmarks and estimated depth. This method does not require 3-D training data, because all the 3-D information is recovered by the symmetry. The symmetric landmarks were labeled by hand. He also tried automatic detecting the symmetric landmarks by neural networks trained from 2-D images, but did not integrate this automatic procedure into the whole framework [24, 25].

2.2. State of the Art in Facial Image Recognition

Several approaches to the overall face identification and recognition problems have been devised over the years with considerable attention paid to methods by which effective face identification and recognition can be replicated [26–28]. The

intuitive way to do face recognition is to look at the major features of the face and compare them to the same features on other faces. Extensive research in face statistical data compression technique using principal component analysis (PCA) technique [29], face features comparisons and recognition based on the Eigenfaces technique [30] and the direct face classification scheme using discrete cosine transform (DCT) coefficients [31, 32] have been reported. In [33], a comparative study of face recognition algorithms has been conducted and several issues such as: *1).* preprocessing and cropping of the faces to equal and consistent sizes, database creation for the preprocessed images; and *2).* a major problem in the amount of light intensity on the original face images have been reported. The comparison of the Eigenfaces algorithm with the Fisherfaces algorithm have also been reported [34, 35]; where it has been recommended that both algorithms can be used for accurate face recognition results except with higher computational load in the later which might limit its application in real-time systems. In addition to the above issues, the face data compression techniques has provided additional challenges to the face identification and recognition problem such as: *1).* generating worse face images which are different from the original image [29], and *2).* the difficulties associated with the restoration of the original face image leading to the inclusion of color information. This second problem has led to the new fractal image encoding research fields [36–38].

In general, the face identification and recognition problem becomes more complicated and computationally intensive as the number of the face images increases especially when used in real-time [39]. As reported in [30], a 90% success in face recognition is achieved using the Eigenfaces algorithm and this algorithm has been widely used in research papers but with greater light intensity on well-taken face images [39–41]. The description of both face detection using the Eigenfaces space and face recognition using neural networks. Real-time face detection from face images was performed in two steps; a normalized skin color map based on the Gaussian function was applied to extract a face candidate region. The facial feature information in the candidate region was employed to detect the face region; face detection was sequentially accomplished using three methods. DFFS, a combination of DFFS and DIFS, and template matching were used. Facial features were extracted according to the Euclidian distance between the determined face region and the pre-defined Eigenfaces from the face region. In the second step, neural network models were trained using 120 images for face recognition. In the experiments, three neural network models corresponding to input variables that included features from face spaces, facial features (geometrical features), and both, were constructed. The image of each person was obtained based on the various directions, poses, and facial expressions. The number of hidden layers was changed from 1 to 3 for several tests of the neural network models. The goal was to reduce lighting effects in order to achieve high-performance of face recognition, because face recognition cannot cope with changes due to lighting, but this recognition algorithm more robust to light variations and requiring less image data has to be [39]. In addition, the variation properties of face images for enhanced optimal identification and recognition by evaluating the statistical data derived from a well-reduced form of each of the original face images under different angular positions and light intensities. Based on the results of their evaluation, they proposed two algorithms based on the: *1).* RBFNN-PCA and *2).* Eigenfaces algorithms, It has also discovered that both algorithm can be used for accurate face recognition results except with high computational load in the later which might limit its application for real time system [18, 42].

In this paper, we examined facial images under different angular positions and light intensities to enhanced optimal identification and recognition by evaluating the statistical data derived from a well-reduced form of each of the original face images with their biodata. Using the results of the evaluation we propose two algorithms based on the: *1).* RBFNN-PCA and *2)* Eigenfaces algorithms, to test for the validity and the efficiency of the two algorithms with higher computational loads for real-time system.

3. Formulation of the Proposed Radial Basis Function Neural Network-Based Neuroscaling Algorithm

3.1. Radial Basis Function Neural Network (RBFNN)

A Radial Basis Function neural network (RBFNN) having the typical structure illustrated in Figure 1 has been proven to be a universal function approximator [43] and an alternative to multilayer perceptron (MLP) feedforward neural network (FNN). The RBFNN is a multidimensional nonlinear function that maps the inputs to the outputs depending on the distance between the input vector and the center vector. Therefore, it can perform similar function mappings as a MLP but its architecture and functionality are very different.

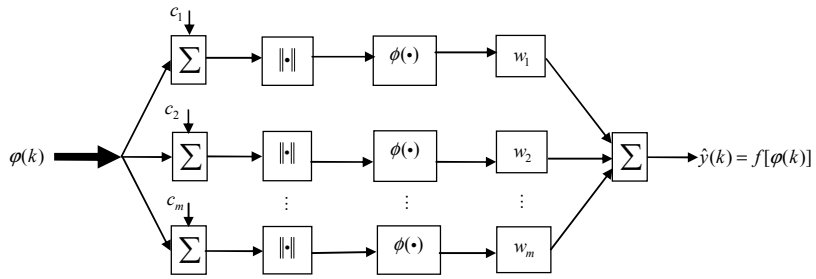


Figure 1. *Radial basis function neural network (RBFNN).*

Consider that the RBFNN shown in Figure 1 with $m-$ dimensional input $\varphi(k) \in \Re^m$ and a single output $\hat{y}(k) \in \Re$ can be represented by the weighted summation of a finite number of radial basis functions as follows:

$$\left.\begin{aligned} &\hat{y}(k) \triangleq f_j[\varphi(k)] = \\ &\sum_{j=1}^{m} w_j \phi\left(\left\|\varphi_j(k) - c_j(k)\right\|\right); \quad j = 1,2,\ldots,m \end{aligned}\right\} \tag{1}$$

where $\phi(\bullet)$ is an arbitrary nonlinear function, $\|\bullet\|$ is the norm that is usually assumed to be Euclidean, $c_j(k) \in \Re^m$ denotes known vectors that represent the centers of the radial basis functions, $w_{j,i}$ is the weight parameter, and $\left\|\varphi_j(k) - c_j(k)\right\|$ is the radial basis function of $\varphi_j(k)$ at time k obtained by shifting $\left\|\varphi_j(k)\right\|$ by $c_j(k)$. However, if the individual elements of the input vector belong to different classes, then a weighted norm can be introduced from [44] as follows:

$$\hat{y}(k) \triangleq f_j[\varphi(k)] = \sum_{j=1}^{m} w_{j,i} \phi_j\left(\left\|\varphi_j(k) - c_j(k)\right\|_{K_j}\right) \tag{2}$$

where $K_i \in \Re^{m \times m}$ is a weight matrix and the weighted Euclidean norm is given by:

$$\left.\begin{aligned} &\left\|\varphi_j(k) - c_j(k)\right\|_{K_j}^2 \\ &\quad = \left\|K_j\left(\varphi_j(k) - c_j(k)\right)\right\|^2 \\ &\quad = \left(\varphi_j(k) - c_j(k)\right) K_j^T K_j \left(\varphi_j(k) - c_j(k)\right) \end{aligned}\right\} \tag{3}$$

The proposed RBFNN is a two layer network that has different types of neurons in the hidden layer and the output layer. The hidden layer, which corresponds to a MLP hidden layer, is a non-linear local mapping. This layer contains radial basis function neurons which most commonly use a Gaussian activation function $g(x)$. These functions are centered over receptive fields. Receptive fields are areas in the input space which activate the local radial basis neurons.

$$g_j[\varphi(k)] = \exp\left[\frac{-\left(\varphi_j(k) - \mu_j\right)^2}{\sigma_j^2}\right] \tag{4}$$

where μ_j is the center of a region called a receptive field, σ_j is the width of the receptive field, and $g_j(\varphi(k))$ is the output of the jth neuron. The basic design method for RBFNNs can be summarized from [45] as: *1).* Random selection of fixed centers, *2).* Self-organized selection of centers, *3).* Supervised selection of centers, and *4).* Regularized interpolation exploiting the connection between the RBF network and the Watson-Nadaraya regression kernel.

The output layer is a layer of standard linear neurons and performs a linear transformation of the hidden node outputs. This layer is equivalent to a linear output layer in a MLP, but the weights are usually solved for using a least square algorithm rather trained for using back-propagation. The output layer may, or may not, contain biases.

Receptive fields center on areas of the input space where input vectors lie, and serve to cluster similar input vectors. If an input vector $(\varphi(k))$ lies near the center of a receptive field (μ), then that hidden node will be activated. If an input vector lies between two receptive field centers, but inside the receptive field width (σ), then the hidden nodes will both be partially activated. When input vectors lie far from all receptive fields there is not any hidden layer activation and the RBF output is equal to the values of the output layer bias.

A radial basis function (RBF) is a local network that is trained in a supervised manner. This contrasts with a MLP network that is a global network. An MLP network performs a global mapping, meaning that all the inputs influences the output, while an RBF network performs local mapping, meaning that only inputs near receptive field produces activation.

The ability to recognize whether an input is near the training set or if it is in an untrained region of the input space gives the RBF a significant benefit over the standard MLP. Since networks generalize improperly and arbitrarily when operating in regions outside the training area, no confidence should be given to their outputs in those regions. When using an MLP, one cannot judge whether or not the input vector comes from these untrained regions and therefore, one cannot judge whether the output contains significant information. On the other hand, an RBF can tell the user if the network is operating outside its training region and the user will know when to ignore the output. This ability makes the RBF the network of choice for safety critical applications or for applications that have a high financial impact.

Designing an RBF neural network requires the selection of the width parameter of the radial basis function. This decision is not required for an MLP. The width should be chosen so that the receptive fields overlap but so that one function does not cover the entire input space. This means that several radial basis neurons have some activation to each input but all radial basis neurons are not highly active for a single input.

Another choice to be made is the number of radial basis neurons. Depending on the training algorithm used to implement the RBF, this may, or may not, be a decision made by the designer. For example, the MATLAB Neural Network Toolbox [46] has two training algorithms. The first algorithm centers a radial basis neuron on each input vector. This leads to an extremely large network for input data composed of many patterns. The second algorithm incrementally adds radial basis neurons to reduce the training error to the preset goal. There are several network architectures that will meet a specified error criterion. These architectures consist of different combinations of the radial basis function widths and the

number of radial basis functions in the network.

The maximum number of neurons is the number of input patterns; the minimum is related to the error tolerance and the complexity of the mapping. This minimum must be experimentally determined. A more complex map and a smaller tolerance require more neurons. The minimum width constant should overlap the input patterns and the maximum should not cover the entire input space. Excessively large widths can sometimes give good results for data with no noise, but these systems usually fail under real world conditions in which noise exists. The reason that the system can train well with noise free cases is that a linear method is used to provide solutions for the second layer weights. The use of a regression method will minimize the error, but usually at the expense of large weights and significant over-fitting. This over-fitting is apparent when there is noise in the system. A smaller width will do a better job of alerting that an input vector is outside the training space, while a larger width may result in a network of smaller size and faster execution time.

3.2. Radial Basis Function Neural Network-Based on Principal Component Analysis

Data visualization is an important means of extracting useful information from large quantities of raw data. The human eye and brain together make a formidable pattern detection tool, but for them to work the data must be represented in a low-dimensional space, usually of two dimensions. Even quite simple relationships can seem very obscure when the data is presented in tabular form, but are often very easy to see by visual inspection. This collects together a number of visualization techniques implemented, both linear and non-linear [47]:

1). Principal Component Analysis (PCA), a classical linear projection method;
2). Probabilistic Principal Component Analysis (PPCA) and mixtures of PPCA, a latent variable formulation of PCA that provides a density model;
3). Generative Topographic Mapping (GTM), a non-linear latent variable model;
4). Neuroscale, a non-linear topographic (i.e. distance preserving) projection that uses an RBF network.

Which method to use on a particular application will depend on whether a density model is required, and what prior information is available about how the data is distributed.

PPCA and GTM are density models; in fact both are Gaussian mixture models with certain constraints. In PPCA, each Gaussian has a covariance matrix which is diagonal along q axes and has a single variance parameter to describe all other variance in each component. This is a good model for data that is piecewise linear. In GTM, the centre of each Gaussian is constrained to lie on a smooth manifold of low dimension (usually 1 or 2) but all Gaussians have a common spherical covariance.

Classical PCA and neuroscale are projection methods for mapping data to a lower dimensional space. PCA does this with a linear transformation that preserves as much data variance as possible. As well as being useful in its own right, as a fast and simple method it is commonly used as a component in other training algorithms or for initializing models (e.g. GTM). Neuroscale is non-linear, but is fast to train on datasets of medium size. Large datasets need to be sub-sampled, since the matrix of inter-point distances required for training is N x N, and will exhaust computer memory for large N.

3.3. Radial Basis Function Neural Network-Principal Component Analysis

Principal Component Analysis (PCA) is the most commonly used feature extraction and visualization technique in practice. There are two reasons for its popularity: it is fast and easy to compute, and it retains maximal information (in the sense of retained variance of projected data) amongst all linear projections.

As a brief mathematical background, suppose that we are trying to map a dataset of vectors x^n *for* $n = 1,..., N$ *in* $V = R^d$ to vectors z^n in $u = R^M$, a subspace of V. For visualization, $M = 2$ so that the vectors z^n can be shown in a scatter plot. We can choose an orthonormal basis $u_1, ..., u_M$ for U and extend this to an orthonormal basis $u_1, ..., u_d$ for V. The orthonormality property implies that

$$u_i^T u_i = \delta_{ij} \tag{5}$$

where δ_{ij} is the Kronecker delta. A vector x can be represented by the vector $(x_1,..., x_d)$, or equivalently:

$$x = \sum_{i=1}^{M} x_i u_i + \sum_{i=M+1}^{d} x_i u_i \tag{6}$$

Now suppose that we project to the M-dimensional space spanned by the first M vectors:

$$z = \sum_{i=1}^{M} x_i u_i + \sum_{i=M+1}^{d} b_i u_i \tag{7}$$

where the b_i are constants. We choose the coefficient b_i and vectors u_i so that the projected vectors z^n best approximate x^n. The quality of the approximation is measured by the residual (or reconstruction) sum of squares error between the two vectors:

$$E = \frac{1}{2} \sum_{n=1}^{N} \left\| x^n - z^n \right\| \tag{8}$$

$$= \frac{1}{2} \sum_{n=1}^{N} \sum_{i=M+1}^{d} \sum_{j=M+1}^{d} \left(x_i^n - b_i \right) \left(x_j^n - b_j \right) u_i^T u_j \tag{9}$$

$$= \frac{1}{2} \sum_{n=1}^{N} \sum_{i=M+1}^{d} \left(x_i^n - b_i \right)^2 \tag{10}$$

since the u_i are orthonormal. Setting the derivative of E with respect to b_i to zero, we obtain

$$b_i = \frac{1}{N}\sum_{n=1}^{N} x_i^n \tag{11}$$

which is the *ith* coordinate of the mean vector x with respect to the coordinate system $u_i, \ldots u_d$. This is equal to $u_i^T x$. It follows that the error term in (10) can be written (after reordering the summations) as

$$E = \frac{1}{2}\sum_{i=M+1}^{d}\sum_{n=1}^{N}\left\{u_i^T\left(x^n - \bar{x}\right)^T\right\}\left\{\left(x^n - \bar{x}\right)u_j\right\} \tag{12}$$

$$= \frac{1}{2}\sum_{i=M+1}^{d} u_i^T \sum u_i \tag{13}$$

where Σ is the covariance matrix of the data. Using Lagrange multipliers, it can be shown that the stationary points of E with respect to the vectors u_i occur at the eigenvectors of $\sum$, so that $\sum u_i = \lambda_i u_i$. Substituting such vectors into (13), we find that the residual error is given by

$$E = \frac{1}{2}\sum_{i=M+1}^{d} \lambda_i \tag{14}$$

Since Σ is symmetric, it has a full set of d eigenvectors, and these can be chosen to be orthonormal as required. Furthermore, as it is a covariance matrix, it is positive semi-definite, so each $\lambda_i > 0$. Hence the minimal error E is achieved choosing the $d - M$ smallest eigenvalue in (14) and we project data onto the space spanned by the eigenvectors corresponding to the largest M eigenvalues. These eigenvectors are called the first M principal components.

It is usual to list the principal components in descending order of eigenvalues; so the first principal component corresponds to the largest eigenvalue and has the largest variance of any linear combination of the original variables.

There is no general technique for deciding how many principal components should be used to represent the data adequately (though a Bayesian approach has been proposed for probabilistic PCA), but a useful heuristic technique is to plot the singular values (or their logarithm) to see if there is a point at which the values level off, or to choose a fraction (often 0.90 or 0.95) of the variance to be retained by computing

$$\frac{\sum_{i=1}^{M}\lambda_i}{\sum_{i=1}^{d}\lambda_i} \tag{15}$$

3.4. Probabilistic Principal Component Analysis (PPCA)

The motivation for the definition of PCA given in (5) is in terms of minimizing the squared reconstruction error

$$E = \frac{1}{2}\sum_{n=1}^{N}\left\|x^n - z^n\right\|^2 \tag{16}$$

between the projected and original vectors. The disadvantage of this approach is that it does not define a generative model: there is no density model or, to put it another way, there is no principled interpretation of the error function E. This was the motivation for developing probabilistic PCA, known as PPCA [48]. A density model offers several advantages:

1). The definition of likelihood allows us to compare this model with other density models in a quantitative way.

2). If PPCA is used to model class-conditional densities, then posterior probabilities of class membership may be calculated.

3). A single PPCA model may be extended to a mixture of PPCA models.

4). Bayesian inference methods may be applied if a suitable prior is chosen.

3.5. Generative Topographic Mapping

We have already seen how a latent variable approach provides a generative model for PCA. However, a single low-dimensional linear space is not a realistic model for many real datasets. The aim of the Generative Topographic Mapping (GTM) is to allow a non-linear transformation from latent space to data space but still make the model computationally tractable. In this approach, the data is modeled by a mixture of Gaussians, in which the centers of the Gaussians are constrained to lie on a lower dimensional manifold. The topographic nature of the mapping comes about because the kernel centers in the data space preserve the structure of the latent space. By careful selection of the form of the non-linear mapping (using an RBFNN), it is possible to train the model using a generalization of the expectation maximization (EM) algorithm.

One of the motivations for developing this model is to provide a principled alternative to the self-organizing map (SOM) algorithm proposed in [49], in which a set of data vectors x_n (n = 1,..., N) in a d-dimensional data space is summarized by a set of reference vectors organized on a lower dimensional sheet: the sheet is generally two-dimensional so that the map can be visualized. This algorithm has been successfully used in many applications, but suffers from a number of drawbacks:

1). The absence of a cost function;

2). No general proof of convergence;

3). The lack of a theoretical basis for choosing training algorithm parameters, such as learning rate and neighbourhood function;

4). The lack of a density model.

These problems arise due to the heuristic nature of the SOM algorithm. GTM overcomes most of these problems; the only drawback is that it is more complicated to program.

3.6. The Proposed Neuroscaling Algorithm

A quite different approach to visualization is based not on a density model, but instead on the concept of data topography. This is assumed to be captured by the inter-point distances, usually measured with a Euclidean metric:

$$d_{ij}^{*} = \left\| x_i - x_j \right\| \tag{17}$$

Each data point *xi £ R^d* is projected (by some function yet to be defined) to a point $y_i \in R^d$. The distance between points y_i and y_j is denoted by d_{ij}, and the quality of the projection is measured by the Sammon stress metric:

$$E_{sam} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j>i}^{N}\left(d_{ij} - d_{ij}^{*}\right)^2 \tag{18}$$

The smaller the stress, the more closely the distances between the yi match the distances in the original data space between the xi, and hence the better preserved is the data structure in the projected space used for visualization.

Note that there is no necessity for using the Euclidean metric to measure the distances (or 'dissimilarities') between data points. Instead, other norms can be used, for example the lp norms

$$\left\| x_i - x_j \right\|_p = \left(\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^p \right)^{1/p} \tag{19}$$

where the Euclidean metric is equivalent to p = 2. However, if the main purpose of the projection is to visualise the data, it is worth remembering that our visual system is highly tuned to discriminating patterns based on Euclidean distances, and so it makes most sense for the metric in the projection space to be Euclidean. If the metrics in the two spaces are not the same, it may be hard to interpret the results since spurious structure may be visible simply because of the properties of the metric [50].

One useful change can be made to the distance measure if some additional dissimilarity information is known about the data. For example, if each data point x_i belongs to a known class C_i, a dissimilarity measure s_{ij} can be defined by

$$s_{ij} = \begin{cases} 0, & if\ i = j \\ 1, & if\ i \neq j \end{cases} \tag{20}$$

This can then be incorporated into the distance measure:

$$\delta_{ij} = (1-\alpha)\delta_{ij}^{*} + \alpha s_{ij} \tag{21}$$

where the parameter $\alpha \in [0,1]$ controls the degree of supervisory information in the mapping. When $\alpha = 0$, this is the original definition (17), while if $\alpha = 1$, there is no distance information and points will be projected using only the class information. Intermediate values of α allow the points to be projected so as to retain the distance structure with extra separation from the classes. This can help to expose inconsistencies such as incorrectly labeled data.

In Sammon's original paper [51], there was no mapping defined between $\mathbb{R}^d$ and $\mathbb{R}^q$. Instead, the points y_i, *for i = 1,..., n* were treated as variables in an optimization problem, with E_{sam} used as the objective function. In this approach, the mapping is defined simply by the lookup table of ordered pairs *(x_i, y_i),* This has a number of disadvantages:

1). The number of parameters in the optimization problem grows linearly with the size of the dataset.

2). The mapping is defined only for the original data points xi.

3). The only way to compute the mapping for a new point x* is to add it to the dataset and re-optimize the stress measure (though most of the yi can be started at their earlier values to cut down on some of the computational cost).

4). As the mapping does not generalize, it is impossible to take a sub-sample of the training data to train the map more efficiently.

To get round these difficulties, the neuroscale model was introduced in [52]. This model defines a nonlinear map $R^d \rightarrow R^d$ using a neural network (preferably a RBFNN as proposed here provided a particularly efficient training algorithm will be available for network training). Unfortunately, both the neuroscale [47] and original Sammon mapping [51] suffer from the fact that the computational demands grow with the square of the number of data points. This is simply because there are *N (N —1)/2* distances included in the sum for E_{sam}. The simplest approach to training the model is simply to compute the partial derivatives *of E = E_{sam}* with respect to the network weights, and then to use a non-linear optimization algorithm to find the optimal weights. Using the chain rule, we can write

$$\frac{\partial E}{\partial \omega_{kr}} = \sum_{i}^{N} \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial \omega_{kr}} \tag{22}$$

By differentiating (18), it is easy to see that

$$\frac{\partial E}{\partial y_i} = -2\sum_{j \neq i}^{N}\left(\frac{d_{ij}^{*} - d_{ij}}{d_{ij}}\right)\left(y_i - y_j\right) \tag{23}$$

while, for the proposed RBFNN, the gradient of y_i with respect to the output network weights is given by the same equation as for the sum-of-squares error

$$\frac{\partial y_i}{\partial \omega_{kr}} = \delta_{ir} z_k \tag{24}$$

where z_k is the output of the *kth* hidden unit, and δ_{ir} is the Kronecker delta. Assuming that the hidden unit centers and widths are chosen to model the input data distribution as in the usual two-stage training process for supervised problems, these derivatives are all that is needed to train an RBFNN to perform a topographic projection.

One advantage of using RBF networks in supervised regression problems with a sum-of-squares error function is that the output layer weights can be trained very efficiently with a single pass matrix pseudo-inverse computation. This advantage is lost here, since the error function contains quartic terms. However, it has been showed in [53] that there is at raining algorithm that is more efficient than using a general non-linear optimizer.

This algorithm, called shadow targets, makes use of the

special form of the error function and the linear dependence of the network output on the output layer weights. It is based on a model trust region approach, analogous to that used in the scaled conjugate gradient algorithm.

We write the output of the network in matrix form as

$$\mathrm{Y} = \Phi \mathrm{W} \tag{25}$$

where Φ is the matrix of hidden unit activations including bias terms with the assumption that the hidden unit parameters have been already determined. Thus,

$$\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \phi_0(x_1) & \cdots & \phi_M(x_1) \\ \phi_0(x_1) & \phi_1(x_2) & \phi_0(x_2) & \cdots & \phi_M(x_2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_0(x_N) & \phi_1(x_N) & \phi_2(x_N) & \cdots & \phi_M(x_N) \end{bmatrix} \tag{26}$$

such that we can express (22) (for output dimension r) in the form

$$\tilde{\mathrm{N}}\mathrm{E} = \Phi^{\mathrm{T}} \mathrm{e}_{\mathrm{r}} \tag{27}$$

where

$$\nabla \mathrm{E} = \left(\frac{\partial E}{\partial \omega_{1r}}, \frac{\partial E}{\partial \omega_{2r}}, \frac{\partial E}{\partial \omega_{3r}}, \ldots, \frac{\partial E}{\partial \omega_{Mr}} \right)^T \tag{28}$$

and

$$\mathrm{e}_r = \left(\frac{\partial E}{\partial y_{1r}}, \frac{\partial E}{\partial y_{2r}}, \frac{\partial E}{\partial y_{3r}}, \ldots, \frac{\partial E}{\partial y_{Mr}} \right)^T \tag{29}$$

Equation (27) is simply an expression of the chain rule for a model whose output is linear in its weights. In a least-squares problem, with error

$$E = \frac{1}{2} \sum_{i=1}^{N} \| y_i - t_i \| \tag{30}$$

where t_i represents an explicit target value, the same equation is valid but with

$$\frac{\partial E}{\partial y_i} = (y_i - t_i) \tag{31}$$

The key idea of the so-called shadow targets algorithm is to use (31) to estimate hypothetical targets $\hat{y}$ [54]:

Table 1. *The proposed RBFNN neuroscaling algorithm for facial image recognition based on the shadow targets algorithm.*

1). Initialize the weights W to small random values of appropriate dimensions. Note that weight initialization is carried in RBFNN when the network is constructed using PPCA which projects the dataset x^n.
2). Initialize η to some small positive value.
3). Calculate $\Phi^{\dagger}$ from (26).
4). Use (34) to compute estimated targets $\hat{\mathrm{y}}_{\mathrm{i}}$ using the MLMA described in [54].
5). Solve for the weigh $\mathrm{W} = \Phi^{\dagger}\hat{\mathrm{T}}$ using (33).
6). Calculate E_{sam} from (18) and compare with previous value.
a) If E_{sam} has increased, set $\eta = \eta \times k_{down}$. Restore previous values of W.
b) If E_{sam} has decreased, set $\eta = \eta \times k_{up}$.
7). If convergence has not been achieved, return to Step *4)* (see [54] and [55] for convergence criteria and their detailed description).

$$\left.\begin{aligned} \hat{y}_i &= y_i - \frac{\partial E}{\partial y_i} \\ &= y_i + 2\sum_{j \neq i} \left(\frac{d_{ij} - d_{ij}^*}{d_{ij}} \right) \left(y_i - y_j \right) \end{aligned}\right\} \tag{32}$$

the last step following from (23). The vectors $\hat{y}$ represent (or shadow) the exact targets for the network that would lead to an identical expression for the weight derivatives in the RBFNN in the least squares regression problem.

For a fixed set of targets, the least squares problem can be solved directly:

$$\mathrm{W} = \Phi^{\dagger}\hat{\mathrm{T}} \tag{33}$$

Of course, for our problem the estimated targets $\hat{y}$ are not fixed, since $\mathrm{dE/dy_i}$ depends on the network weights. A logical approach is to iterate this procedure, re-estimating the shadow targets at each step. However, in the early stages of training, the targets estimated by (32) may be poor, and hence the weights given by (33) may increase the error. It is therefore more practical to trust the approximation given by (32) to a limited extent and to increase our trust only when E_{sam} decreases. This is achieved by introducing an additional parameter $\mathrm{r_j}$ and estimating the targets by

$$\hat{\mathrm{y}}_{\mathrm{i}} = \mathrm{y}_{\mathrm{i}} - \eta \frac{\partial E}{\partial \mathrm{y}_{\mathrm{i}}} \tag{34}$$

It is clear that η should be restricted to the range (0, 1). This could pose post robustness and convergence problems. Rather the modified Levenberg-Marquardt algorithm (MLMA) from [55] is adopted in this paper for the minimization of (34). The complete proposed RBFNN neuroscaling algorithm for efficient facial image recognition is summarized in Table 1.

Note that one of the most computationally expensive part of this algorithm is the calculation of the pseudo-inverse $\Phi^{\dagger}$ which must be done only once. The algorithm is robust to changes in k_{down} and k_{up}: in practice the values 0.1 and 1.2 have worked well and has been adopted in this work. It has been shown that the stationary points of this algorithm are the same as gradient-based optimization algorithms, but that it tends to produce models with better generalization [56].

4. Materials, Methods, Implementation Evaluation, Analysis and Discussion of Results

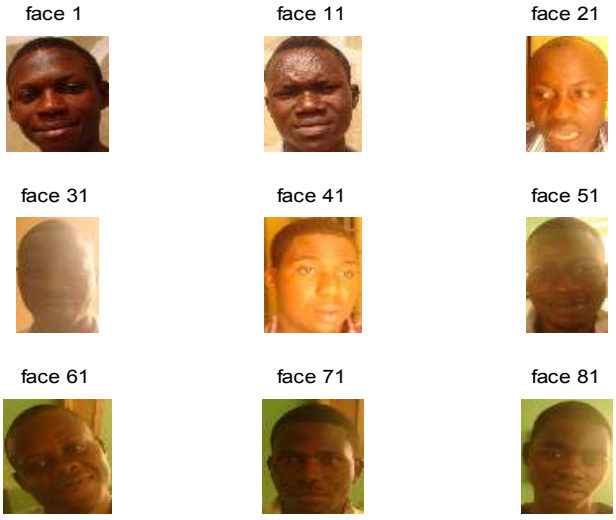
4.1. Materials and Methodology for Obtaining the Faces

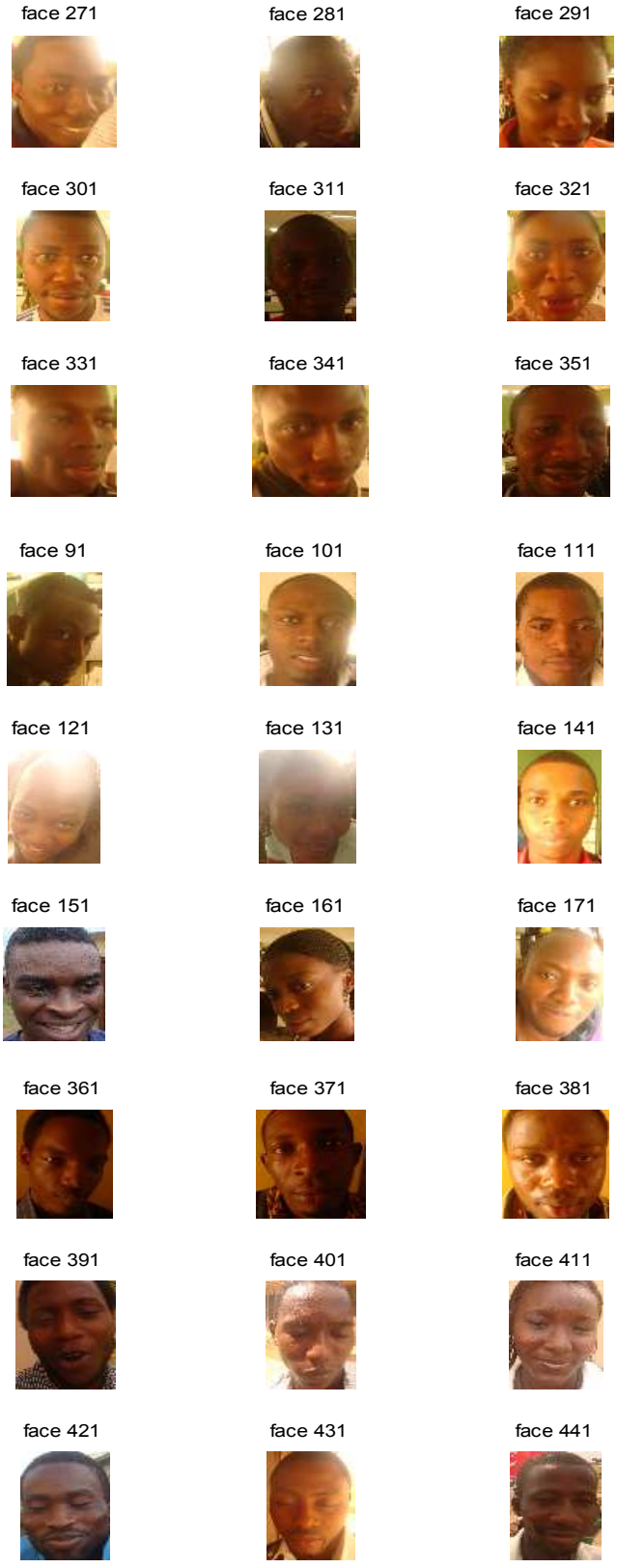
The materials employed in this study were one thousand joint photographic expert group (JPG) face image sample (taken with a Sony cyber shot 10 mega pixel digital camera). The 1000 face image sample consist of 10 different face

sample of 100 individual (student) taken at The Federal University of Technology Akure, Ondo State, Nigeria with their biodata (surname, other names, blood group, colour of eye, hair, mother maiden name and date of birth). Out of the 1000 face Image sample, the first 80 per cent (800 face images) were used to form the database of known faces images while the remaining 20 per cent (200 face images) were reserved for investigating their performance and validation of the proposed face recognition of the 100 individual are shown here. The proposed evaluation algorithm in this study are written, complied and analyzed using MATLAB & Simulink® R2012a [46] software running on an Inter® Core™ Duo CPU E6750@2.67GHz computer on windows™ XP platform.

Pre-processing of the Face Images

The Pre-processing of images is necessary to enhance fast and accurate image identification and recognition. Moreover, when the numbers face images are relatively large, each face image must be pre–processed by reducing its size and converting the image to a format requiring less memory storage capacity. In other to reduce the memory consumption for storing the face images, all the JPG face images were cropped to focus on the whole, converted and resized to portable gray map (PGM) images of height 114 pixels and width 114 pixels, filtered to remove noise using adaptive Wiener filter (AWF). The filtered and resized face PGM images were then used for analysis in other to identify and recognized any face image within the database with their respective biodata. The pre-processed face images (PPFI) used for analysis are shown in Figure 2 and 3 (note that rather than presenting all the 1000 face image sample used in this work, only the first out of the ten face images for each 10 of the 100 individual is shown in Figures 2 and 3 for space economy). The proposed algorithms and Eigenfaces was implemented on the pre-processed face images (PPFI) as shown in Table 2.





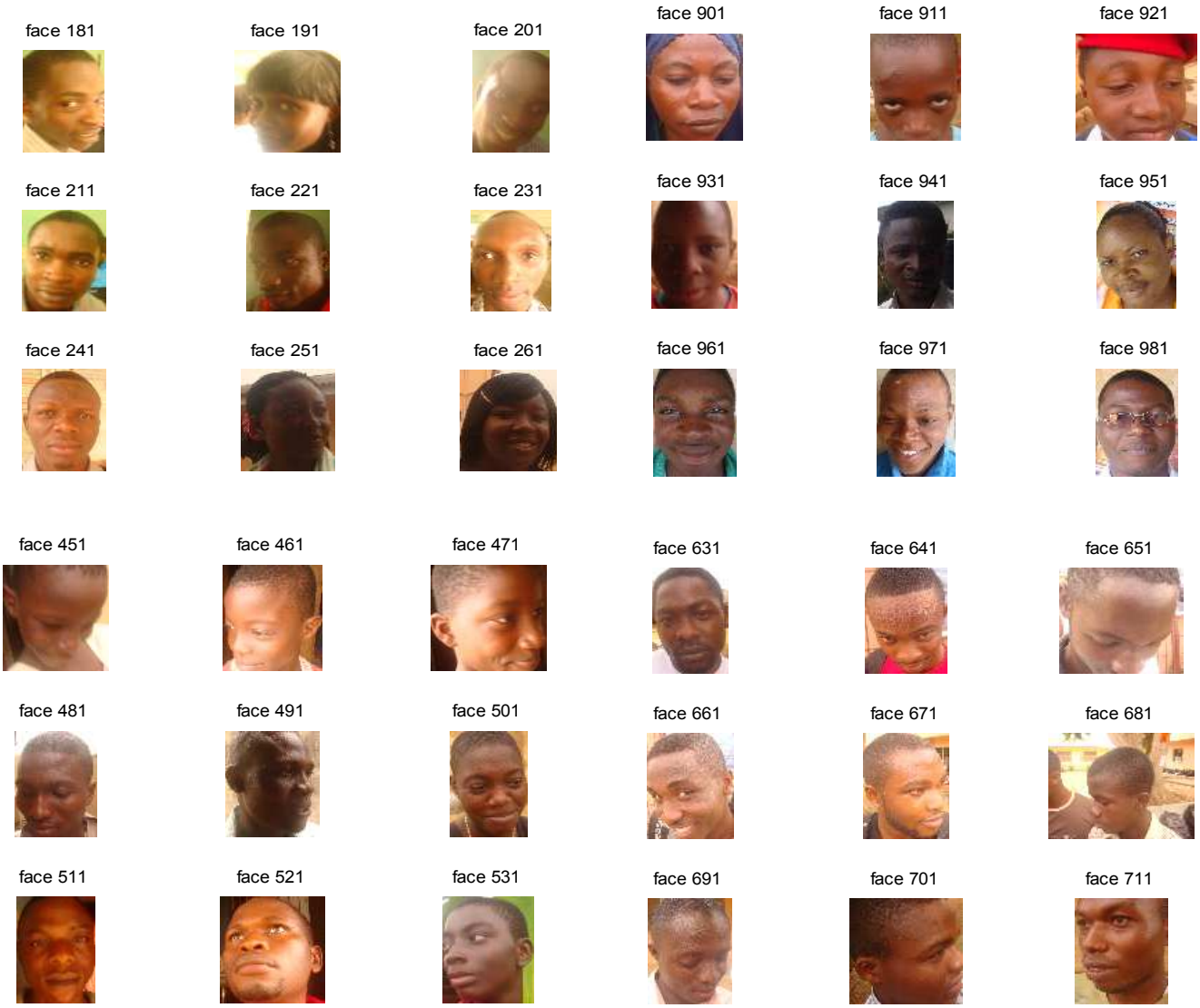


Figure 2. *The pre-processed facial images (PPFI) with AWF for facial images from 1 to 540. Only the first out of ten facial images is shown here for space economy.*

Figure 3. *The pre-processed facial images (PPFI) with AWF for facial images from 541 to 1000. Only the first out of ten facial images is shown here for space economy.*

Table 2. *Implementation and evaluation of the proposed RBFNN neuroscaling algorithm for facial image recognition using MATLAB®.*

```
% 1. Compute the time required for the implementation of the algorithm
start_time_1 = cputime;
% 2.<<<< Load the database information (Only once for faster subsequent
recognition) >>>>
% Load the known face images from database (Only once for faster
subsequent comparison)
% load o1_pre_pro_image
% load o2_pca_anal
disp (' ')
database_time_1 = cputime - start_time_1
% 3.>>>> Create the backgroung colour for inserting images >>>>
figure ('color','black');
% 4.>>>> Select the particular face image to be recognized >>>>
src_face = input ('Please enter any number between 1 and 1000 to recognize
image = ');
% 5.<<<< Evaluate the image correspon64ding the input numerical
identity >>>>
% Evaluate the input face image identity number to check if it is within
% the database limit. Otherwise prompt the user to enter a valid identity
number.
o4_eval_src_face;
% 6.>>>> Implement the Eigenface Algorithm <<<<
start_time_2 = cputime;
o5_face_recog;
disp (' ')
eigface_time_2 = cputime - start_time_2
% 7.>>>> Implement the RBFNN-PPCA Neuroscaling Algorithm >>>>
% start_time_3 = cputime;
o7_rbfnn_neuro_eval;
disp (' ')
rbfnn_neuro_time_3 = cputime - start_time_3
% 8. Biodata for the rcognied face image.
o9_face_biodata;
```

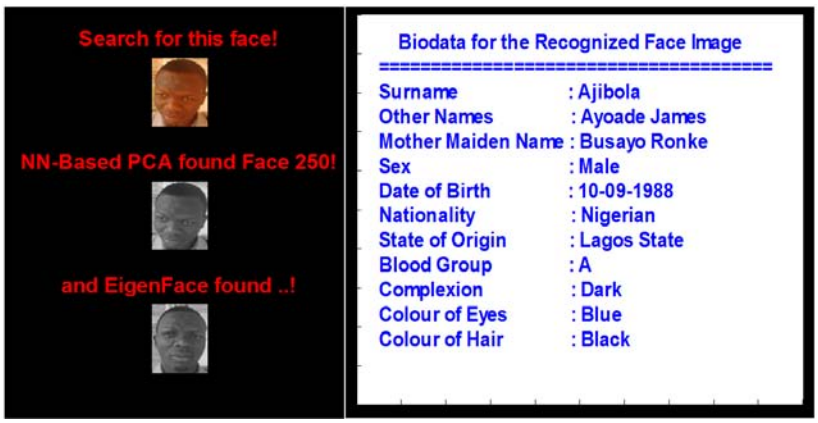
Table 2 shows the face image evaluation algorithm, the first line algorithm in the table above computes the time taken for the Radial Basis Function Neural Network Principal Component Analysis, Eigenfaces and load images of the data base, after which a background colour is created for the image. In the process, a valid number is selected between 1 and 1000, it then evaluates the number to see if it falls between the database. The Eigenfaces and the Radial Basis Function Neural Network Principal Component Analysis algorithm are implemented, alongside with the biodata and the output are displayed respectively.

4.2. Implementation of the Radial Basis Function Neural Network-Based Probabilistic Principal Component and Eigenfaces Algorithms for INFI Recognition

The RBFNN-PPCA and the Eigenfaces recognition algorithms in the present study is implemented in such a way that the user of the algorithm is prompted for the INFI identify as a numerical number between 1 and 1000 which corresponds to the number of face images in the database with their corresponding biodata.

If any numerical value between 1 and 800 is supplied, the RBFNN-PPCA and the Eigenfaces recognition algorithms are activated. Initially, the Radial basis function neutral network-based principal components (PCs) of all the known 800 face images in the database were computed and stored in a database. The PC of the INFI is then computed and compared with the known 800 face image PCs in the database with their corresponding biodata. If the PC of the INFI corresponds to any PC in the PC database, the corresponding face image and the biodata of the INFI are displayed as shown of Figure 4 (a), (b), and (c). However, if there is no corresponding PC for the INFI in the PC database, the comments in Figure 5 (a) and (b) are displayed (instead of comments similar to those shown in Figure 4) indicating non-existence of such face image in the database, On the other hand, the comments shown Figure 4 were obtained for INFI numerical values ranging between 801 and 1000 which correspond to the 200 unknown face images (i. e., the last 20% of the 1000 face image samples) However, when numerical values outside the ranges 1 to 1000 are used, the proposed algorithm and Eigenfaces responds with the comments shown in Figure 6 and terminates automatically.

The Eigenfaces algorithm is computationally intensive, when compared to the RBFNN-PCA, the database load time is insignificant compare to the time taken by the Eigenfaces algorithm, and the position of the images in the database has no effect on the time on both algorithms. The database loading time is suppressed i. e. database of known face images are loaded once, so that only the time of the RBFNN-PCA and Eigenfaces algorithms are computed as shown in Table 3. Figures 7 (a), (b) and (c) shows the efficiency of RBFNN-PCA over Eigenfaces which made the proposed algorithm a better algorithm in speed (time) and accuracy for image recognition in real-time identification.

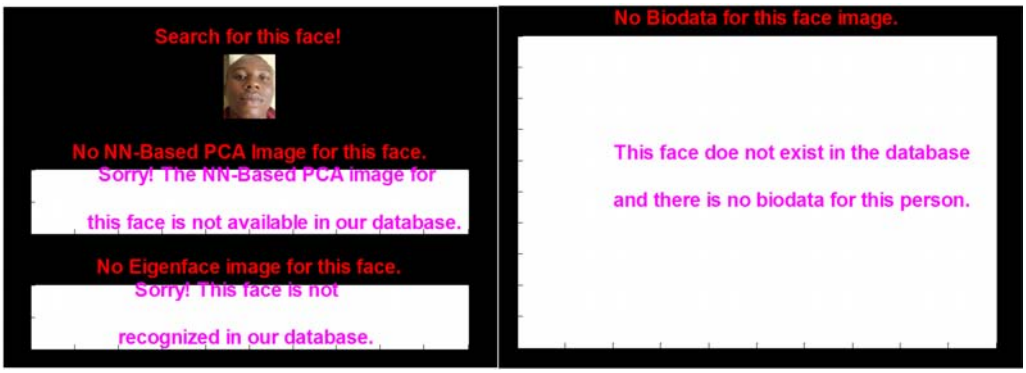


(a)

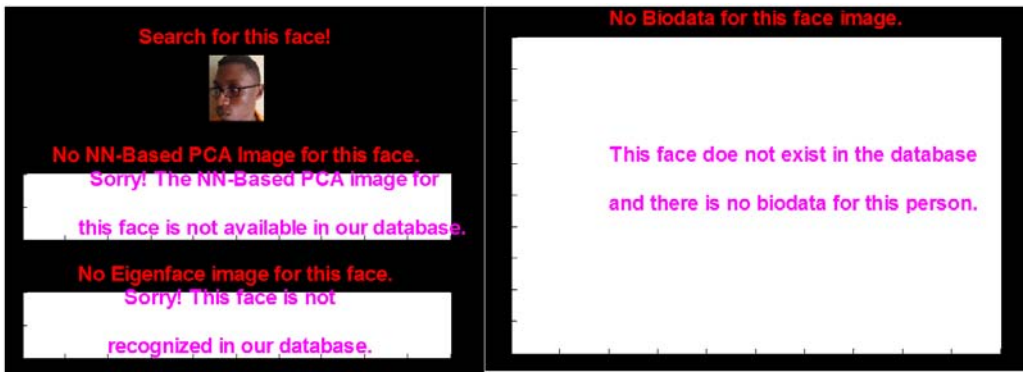
(b)

(c)

Figure 4. *The results obtained when the incoming face (INFI) with non-identity is recognized by both RBFNN-PPCA-neuroscaling and eigenfaces algorithms with their respective biodata.*



(a)



(b)

Figure 5. *(a) The results obtained when the incoming face images (INFI) is not part of the known and unknown database, and (b) The results obtained when the incoming face images (INFI) is not part of the known and unknown database.*

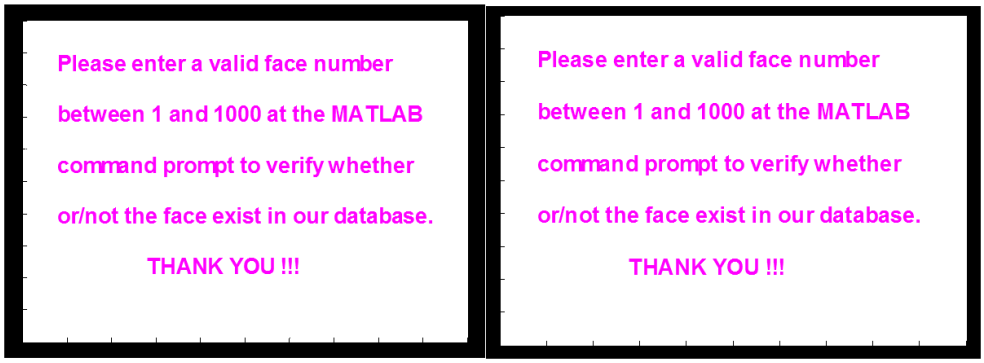


Figure 6. *The results obtained when the face number of the incoming face images (INFI) is outside the range of known facial image.*

Table 3. *Comparison of the computation time for database initialization as well as the computation time required for the implementation of the proposed RBFNN neuroscaling algorithm for facial image recognition.*

INFI number	Database initialization time	Computation time for Eigenfaces algorithm implementation	Computation time for RBFNN Neuroscaling algorithm implementation
Face 500	4.2588	16.3021	0.1716
Face 749	0	16.5517	0.1716
Face 101	0	16.7233	0.0936

5. Conclusion and Future Direction

The radial basis function neural network-based probabilistic principal component analysis (RBFNN-PPCA) has efficient and enhanced optimal face recognition. This algorithm has been tested and validated with 1000 face image samples stored in a database (i. e. 800 known and 200 unknown face images), and Eigenfaces algorithm.

The performance of the proposed algorithm on the 1000 face image samples taken under varying angular positions and light intensity conditions demonstrates the efficiency and adaptability of the proposed algorithm in critical and unprecedented situations.

(a)

(b)

(c)

Figure 7. *The results showing the efficiency of the proposed RBFNN-PPCA neuroscaling algorithm over the Eigenface algorithm.*

The RBFNN-PPCA neuroscaling algorithm for facial image recognition (i. e. the prompted numerical values) can be likened to bank account, finger print, iris, signature, identity card or passport numbers as a means of personal identification to checkmate crimes and fraudulent activities. The speed and accuracy of the proposed algorithm based on the implementation strategy show that it could be adapted for online face analysis and recognition in real-time.

It should implemented on real-time embedded reconfigurable computing machines such as digital signal processors (DSPs), field programmable gate arrays (FPGAs), complex programmable logic devices (CPLDs) as well as other dedicated high performance computing platforms with real-time camera interfaced for on-line facial image detection, identification, recognition, tracking and analysis systems design and deployment.

References

[1] M. Bryant, "Ten Great Uses of Face Recognition", Retrieved 19 August, 2015, Available [Online]: http://www.thenextweb.com.

[2] A. Gupta, S. Satkin, A. A. Efros and M. Hebert, "*From 3- Scene Geometry to Human Workspace*", The Robotics Institute, Carnegie Mellon University, 1961.

[3] A. J. Goldstein, L. D. Harmon and A. B. Lesk, "Identification of human faces", *In Proceedings of IEEE. May 1971*, vol. 59, no 5, 748–760, 1971.

[4] C. E. Chapel, "Fingerprinting: A manual of identification", 1941.

[5] SONY, "Cyber-shot Compact Digital Cameras", Sony, 2016. Available [Online]: http://store.sony/.../S_Digital_camera.com.

[6] F. R. Cherrill, "The Fingerprint system", Scotland Yard (1954).

[7] J. A. Markowitz, "Voice Biometrics", *Communications of the ACM*, vol. 43, no. 9, pp. 66, 2000.

[8] R. L. Zunkel, "Hand Geometry Based Verification", Springer-Verlag, London, 2008.

[9] C. A. Sebastopol, O'Reilly & Associates, Chapter 3, 2000.

[10] P. J. Phillips, H. M. Syed, A. Rizvc and P, J. Rauss, "The FERET evaluation methodology for face recognition algorithms", *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 22, pp. 1090–1104, 2000.

[11] T. Kwon and J. Lee, "Practical Digital Signature Generation using Biometrics", Springer, 2010.

[12] MIT, "Why Face Recognition?", 016, Available [Online]: http://vismod.media.mit.edu/tech-reports/TR-516/node2.html.

[13] G. W. Wilton, "Fingerprints: History, law and Romance", 1938.

[14] F. Hughes, D. Lichter, R. Oswalo and M. Whitfield, "Face Biometrics: A longitudinal study", 2006.

[15] T. Ahonen, A. Hadid and M. Pietik, "Face Recognition with Local Binary Patterns", pp. 469–480, 2004.

[16] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 17, pp. 711–720, 1997.

[17] S. Bentin, T. Allison, A. Puce, E. Perez and G. McCarthy, "Electrophysiological studies of face perception in humans", *Journal of Cognitive Neuroscience*, vol. 8, no. 6, pp. 551–565, 1996.

[18] Y. Chan, S. H. Lin and S. Y. Kung, (1998): Video Indexing and Retrieval", *Multimedia Technology for Applications*, Sheu and Ismail editors, IEEE Press, 1998.

[19] V. A. Akpan and R. A. O. Osakwe, "Face Image Processing, Analysis and Recognition Algorithms for Enhanced Optimal Face Recognition Systems Design: A Comparative Study", *African Journal of Computing & ICT,* vol. 2, no. 2, pp. 21–40, 2009.

[20] V. Bavel, "Biometrics at the Frontiers: Assessing the Impact on Society", February 2005.

[21] Y. Wang, T. Tan, and A. K. Jain, "Combining Face and Iris Biometrics for Identity Verification", *In Proceedings of the 4th International Conference on AVBPA*, June 2003.

[22] P. H. Lee, G. S. Hsu, T. Chen and Y. P. Hung, "Facial trait code and its application to face recognition", 2008.

[23] D. Lay, *Linear Algebra and It's Applications*. New York: Addison-Wesley, 2000.

[24] L. C. Jain, "Intelligent Biometric Techniques in Face Recognition", CRC Press, U. S. A., 1999.

[25] A. K. Jain, P. J. Flynn and A. Poss, "*Handbook of Biometrics*", Springer–Verlag, London, 2007.

[26] J. Wu, W. A. P. Smith and E. R. Hancock, "Facial gender classification using shape from shading," *Image and Vision Computing*, doi: 10.1016/j.imavis.2009.09.003, 2009.

[27] G. C. Littlewort, M. S. Bartlett and K. Lee, "Automatic coding of facial expressions displayed during posed and genuine pain," *Image and Vision Computing*, vol. 27, pp. 1797–1803, 2009.

[28] S. Lucey, Y. Wang, M. Cox, M. Sridharan and J. F. Cohn, "Efficient constrained local model fitting for non-rigid face alignment," *Image and Vision Computing*, vol. 27, pp. 1804–1813, 2009.

[29] L. I. Smith. (Feb. 26, 2002). "A Tutorial on Principal Component Analysis", *Technical Report"*, Available [Online]: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

[30] M. Robinson, M. Escarra, J. Kruegerand and D. Kochelek, "Face Recognition using Eigenfacess", *Course Number: ELEC 301*, CONNEXIONS, Rice University, Houston, Texas, 2009. Available [Online]: http://cnx.org/content/col10254/1.2/.

[31] D. J. Duh, J. H. Jeng and S. Y. Chen, "DCT based simple classification scheme for fractal image compression," *Image and Vision Computing*, vol. 23, pp. 1115–1121, 2005.

[32] N. Chen, K. Chung and J. Hung, "Novel fractal encoding algorithm using normalized one-norm and kick-out condition", *Image and Vision Computing,* (2009), doi: 10.1016/j.imavis.2009.08.007.

[33] P. K. Yalamanchili and B. D. Paladugu, "Comparative Study of Face Recognition Algorithms", Final Project, ECE847: Digital Image Processing, Department of Electrical Engineering, Clemson University, South Carolina, U. S. A., 2007. Available [Online]: http://www.ces.clemson.edu/~stb/ece847/fall2007/projects/face%20recognition.pdf.

[34] P. N. Belhunmeur, J. P. Hespanha and D. J. Kriegman, "Eigenfacess vs. Fisherfaces: recognition using class specific linear projection", *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711–720, 1997.

[35] J. Choi, S. Lee, C. Lee and J. Yi, "A real-time face recognition system using multiple mean faces and dual mode fisherfaces", *IEEE International Symposium on Industrial Electronics (ISIE 2001)*, pp. 1686–1689, 2001.

[36] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations", *IEEE Trans. Image Process*, vol. 1, no. 1, pp. 18–30, 1992.

[37] T. K. Truong, J. H. Jeng, I. S. Reed, P. C. Lee and A. Q. Li, "A fast encoding algorithm for fractal image compression using the DCT inner product", *IEEE Trans. Image Process*, vol. 4, no. 4, pp. 529–535, 2000.

[38] R. Distasi, M. Nappi and D. Riccio, "A range/domain approximation error-based approach for fractal image compression", *IEEE Trans. Image Process*, vol. 15, no.1, pp. 89–97, 2006.

[39] H. Bae and S. Kim, "Real-time face detection and recognition using hybrid information extracted from face space and facial features", *Image and Vision Computing*, vol. 23, pp. 1181–1191, 2005.

[40] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neurosicence*, vol. 3, no. 1, pp. 71–86, 1991.

[41] M. Turk and A. Pentland, "Face recognition using Eigenfaces," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586–591, 1991.

[42] L. Yi-Shin, N. Wai-Seng and L. Chun-Wei, "*A Comparison of Different Face Recognition Algorithms*", Technical Report, National Taiwan University, 2009.

[43] J. Park and L. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks" *Neural Computation*, vol. 3, pp. 246-257, 1991.

[44] M. M. Gupta, L. Jin and N. Homma, "Static and Dynamic Neural Networks: From Fundamental to Advanced Theory". *Hoboken, New Jersey: John Wiley & Sons*, 2003.

[45] Haykin, S. (1999). "Neural Networks: A Comprehensive Foundation". 2nd ed. *Upper Saddle River, NJ: Prentice-Hall*.

[46] The Math Works Inc., MATLAB & Simulink® R2012a, 3 Maple Drive, California, U. S. A. http://www.mathworks.com.

[47] I. T. Nabney, "Netlab: Algorithm for Pattern Recognition", Springer-Verlag, London, 2004.

[48] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis", J. Roy. Statist Soc. B 61, 611–622, 1999.

[49] T. Kohonen, "Self-organized formation of topologically correct feature maps", Biological Cybernetics 43, 59–69, 1982.

[50] N. P. Hughes, "Artefactual Structure from Topographic Mappings", Master's thesis, Aston University, Birmingham, United Kingdom, 1999.

[51] J. W. Sammon, "A nonlinear mapping for data structure analysis", *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.

[52] D. Lowe and M. E. Tipping, "Feed-forward neural networks and topographic mappings for exploratory data analysis", *Neural Computing and Applications*, 4, 83 – 95, 1996.

[53] M. E. Tipping and D. Lowe, "Shadow targets: A novel algorithm for topographic projections by radial basis functions", *In Proceedings of the IEEE International Conference on Artificial Neural Networks*, Vol. 440, pp. 7–12, 1997.

[54] V. A. Akpan and G. D. Hassapis, Training dynamic feedforward neural networks for online nonlinear model identification and control applications, *International Reviews of Automatic Control: Theory & Applications*, vol. 4, no. 3, pp. 335–350, 2011.

[55] V. A. Akpan, "*Development of new model adaptive predictive control algorithms and their implementation on real-time embedded systems"*, Ph. D. Dissertation, 517 pages, 2011. [Online] Available: http://invenio.lib.auth.gr/record/127274/files/GRI-2011-7292.pdf.

[56] M. E. Tipping, "*Topographic Mappings and Feed-Forward Neural Networks*", Doctoral (Ph.D) thesis, Aston University, Birmingham, United Kingdom, 1996.

Biography

Vincent A. Akpan holds a Ph. D. degree in Electrical & Computer Engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece in 2011. He is currently a Senior Lecturer and the Head of Biomedical Technology Department, FUTA, Akure, Nigeria. His research interest is in advanced instrumentation, intelligent control, embedded systems & Rehabilitation engineering. He is the co-author of a book and has authored and/or co-authored more than 50 articles in refereed journals and conference proceedings. Dr. Akpan is a member of The IEEE, USA; The IET, UK; The IoP, UK; a Fellow of the College of Biomedical Engineering & Technology (CBET), Nigeria; and a Fellow of the IPMD, Nigeria.

Joshua B. Agbogun received an M. Sc. degree in Computer Science from University of Nigeria Nsukka (UNN), Enugu State, Nigeria in 2010. He is currently Lecturer II with the Department of Mathematical Sciences, Kogi State University, Anyigba, Nigeria. He is currently working towards his Ph. D at Kogi State University, Anyigba. His integrated research interests include: Deep Web as a search tool and Development of Machine Learning Algorithm for Information Extraction from semi Structured Data. He is the author of a book and has authored and/or co-authored 9 articles in refereed Journals and conference proceedings. Mr. Agbogun is a member of Nigeria Computer Society (NCS) since 2010 and currently the Chairman of NCS Kogi State Chapter.

Michael T. Babalola obtained a Ph. D in Communication Physics from the University of Ibadan, Nigeria in 1976. He is currently a Professor and the Head of Physics Electronics Department with Afe Babalola University, Ado-Ekiti, Nigeria. His research interests are on electronic measurement and instrumentation, computer systems architecture, physics of the lower and upper atmospheres, system identification and adaptive control. He has authored and/or co-authored more than 60 articles in refereed journals and conference proceedings. Professor Babalola is a member of the SAN, Nigeria and the NIP, Nigeria.

Bamidele A. Oluwade obtained his Ph. D degree in 2004 in Computer Science from the University of Ibadan, Nigeria. He is currently a Professor and the Head of Department of Computer Science with the University of Ilorin, Iloring, Nigeria. He has authored and/or co-authored more than 80 articles in refereed journals and conference proceedings including book chapters. Professor Oluwade is a member of the Solar Energy Society of Nigeria; Fourth Dimension Science Society, Nigeria; Sigma Xi, The Scientific Research Society, USA; Nigerian Mathematical Society; Institute of Electrical and Electronics Engineers (IEEE), U. S. A.; Nigeria Computer Society, American Mathematical Society, USA; International Federation of Nonlinear Analysts, IEEE Communications Society. USA.