International Journal of Innovative
Computing, Information and Control
Volume 9, Number 7, July 2013

ICIC International ©2013 ISSN 1349-4198
pp. 2873–2886

PERFORMANCE EVALUATION FOR FACE IDENTIFICATION ALGORITHM USING Z-BASED SPECIFICATION AND EMPIRICAL ANALYSIS

Gulzar Ali Khuwaja[1] and Nazir Ahmad Zafar[2]

[1]Department of Computer Engineering
[2]Department of Computer Science
College of Computer Sciences and Information Technology
King Faisal University
P O Box 400, Al Ahsa 31982, Saudi Arabia
{ khuwaja; nazafar }@kfu.edu.sa

Received June 2012; revised October 2012

Abstract. *Face identification is an important issue that has gained much importance in the area of image processing. Normally testing is carried through simulation in pattern recognition but unfortunately it is lacking in verifying correctness of the models. In addition, the number of simulations increases exponentially to cause state space explosion problem and regression testing is required to re-conduct the entire simulation in case of any modification in a system. Validation and verification are processes that help to ensure that models and simulations are correct and reliable. The Z/Eves tool is used because it is a powerful one for specification analysis and provides various exploration techniques to prove the properties of a system. We focus on two aspects in this paper: i) automatic identification of faces using adaptive and invariant neural network architecture and ii) evaluating performance for an identification algorithm based on formal specification. The architecture is applied to show its feasibility using a collection of 890 faces of 110 subjects from two standard databases, ORL and PICS. Empirical results yield a recognition accuracy rate of 96.7% on 40 subjects and 240 face images for ORL.*
Keywords: Adaptive classification, Face identification, Computer vision, Formal specification, Performance evaluation

1. **Introduction.** Invariant face classification is a challenging task, especially in the absence of highly controlled environments and recognition constraints. Although the process for recognizing faces has received considerable attention in the course of computer vision research [1,2], the present stage is far from the goal of human-like capability especially from the point of robustness to wide-range lighting. The reason is that daily shadow, reflection, and darkness have great influence on the accuracy of facial recognition through the inevitable change of gray level.

Pattern variability often causes poor pattern recognition accuracy. Earlier publications reported findings on pattern recognition based on non-adaptive feature fusion [3] and decision fusion [4]. The reported investigations showed that the use of multiple information sources, exhibiting redundancy or complementarity, do not itself result in robust pattern recognition. The studies showed that mismatched recognition and fusion-training conditions lead to poorer recognition accuracy than matched conditions. This suggests that robust pattern recognition may require adaptation [5].

It is quite challenging to find reliable scheme for person verification for small and large number of objects. The task to perform, in addition to the recognition, is the extreme subjectivity of performance of algorithms on various databases. Unless we have one universal face database which can be used as benchmark for testing performance of various algorithms and scheme for face verification, objectively evaluating the performance of algorithm becomes very difficult [6].

In most of the work mentioned above, testing is carried through simulation in pattern recognition but unfortunately this approach is lacking in verifying the correctness of such systems. Further, the number of simulations increases exponentially to provide a required level of confidence due to their complexity. In addition, when a modification to the system is needed, regression testing will be required to perform which suggests that the complete set of simulations must be re-conducted. The most important is that the performance promises obtained through simulations are not absolute even if they do provide a conditional proof that the system is correct under certain assumptions. To overcome the issues of simulation, validation and verification are processes that help to ensure that models and simulations are correct and reliable. Therefore, it has become indispensable to apply advanced techniques, for example, formal approaches which provide an exhaustive support for verification of algorithms before the simulation. Performance evaluation is becoming a critical aspect of emerging systems. As a result, the use of dominant performance modeling techniques becomes of primary significance. The performance evaluation objective is to gain some opinion of how the system will perform under assumed situations. When systems become more complicated and safety and/or security become a real issue, formal approaches to system design offer another level of protection.

Formal description techniques describe the structure and the behavior of systems. Their objective is to produce correct and unambiguous descriptions, which should ultimately produce efficient and error free implementations. The input to the formal specification is a set of requirement statements and a conceptual overview of the proposed system. The output is a formal representation that captures the functional description of the proposed system [7]. Formal methods differ from other design systems through the use of formal verification schemes. The basic principles of the system must be proven correct before they are accepted [8]. Despite the differences over the applications of formal methods their use in complex systems is increasing. It is very important to note that formal verification does not remove the need for testing [9]. This is because formal verification cannot resolve the issue of unrealistic assumptions that are made in the design but it can help detect errors in reasoning which would otherwise be left unproven.

This work addresses two aspects by using Z-based formal specification and simulation techniques: i) automatic identification of human faces using an adaptive neural network architecture that is trained on small-sized and low-resolution images and ii) evaluating performance for an identification algorithm based on formal specification. The neural network has shown its feasibility using a collection of 400 face images from the ORL database [10] and 490 face images from the PICS database [11]. Formal specification and simulation are combined in this research because both have always been complementary to system design by bridging the gap between these approaches. Simulation validates the behavior of a system for a particular computation while verification provides an entire set of computations [12]. Further, traditional system design has used extensive testing to verify behavior, but testing is capable of only finite conclusions. It has been demonstrated that tests can only show the situations where a system will not fail, but cannot say anything about the behavior of the system outside of the testing scenarios [13]. On the contrary if a theorem is proven true it remains true.

Formal analysis is an effective technique for describing properties to evaluate the correctness of systems, alleviating the shortcomings uncovered through exhaustive testing and simulation. At basic level of application of formal approaches, formal specification may be described and then program can be developed or generated in an informal or automated way. This is assumed as most cost-effective option in applications of formal methods. Z notation is a model oriented specification language based on set theory and first order predicate logic and is used for verification of the adaptive neural network based face identification algorithm underhand [14]. For the reason that it is based upon set theory including standard set operators which are required for defining complex structures, for example, matrices and operations over it. The Z/Eves tool is used because it is a powerful one for specification analysis and provides various exploration techniques to prove the properties of a system [15].

Rest of the paper is organized as follows: Related work is given in Section 2. Section 3 presents the databases used for this work and the preprocessing of face images. The algorithm development for the architecture and its performance evaluation based on formal specification are described in Section 4. The empirical results are presented and discussed in Section 5. Finally, in Section 6, the paper outcomes are concluded.

2. **Related Work.** Face identification is one of the topics in computer vision and pattern recognition that has gained much attention of researchers from academia and industry. A texture based face representation and verification method called Log-Gabor Mixture Model is introduced in [6]. The main advantage of this method is that it gives more than one set of features unlike the standard Log-Gabor filter. Another method for high-speed face identification based on the discrete cosine transform (DCT), the Fisher's linear discriminant (FLD) and radial basis function (RBF) neural networks is presented in [16]. A new technique based on linear programming for feature selection and classifier training is introduced in [17]. The task of face expression recognition with a small number of training images of an expression is also considered. In [18], an IPCA method based on the idea of singular value decomposition (SVD) updating algorithm, namely an SVD updating-based IPCA (SVDU-IPCA) algorithm is proposed and has been evaluated using available public databases, namely FERET, AR, and Yale B. A new classifier called total margin-based adaptive fuzzy support vector machines (TAF-SVM) dealing with several problems that may occur in support vector machines (SVMs) when applied to the face identification is presented in [19]. A general classification algorithm for image-based object recognition based on a sparse representation computed by l-minimization is proposed in [20]. In [21], a constrained optical flow algorithm, which combines the advantages of the unambiguous correspondence of feature point labeling and the flexible representation of optical flow computation, has been developed for face recognition from expressional face images. A method for recognizing faces degraded by blur using deblurring of facial images is proposed [22] and experiments on a large face database (FERET) artificially degraded by focus or motion blur have shown that it substantially improves the recognition performance compared to existing methods. At the onset of learning, one does not know the number of clusters in the data set. Normally, therefore, this number is overestimated by including excess units or neurons in the network [23]. This means that after convergence, some neurons will be redundant in the sense that they do not evolve significantly and thus do not capture any data clusters. Typically, these neurons are initialized to points in the weight space that have relatively low overlap with the training data points. By visualizing the learned pattern of the hidden layer neurons, it is found that there are neurons with completely blurred patterns called blind neurons [24]. These neurons do not see the faces which are clamped to the neurons of the input layer. Survival of the fittest neurons are then crucial to the progress of the learning process. Eliminating the blind neurons enhances the classifier performance both on time and space. The number of neurons is taken as the minimum possible class to shorten the training and test time of the classifier.

3. **Face Databases and Preprocessing.** In practice, published recognition results are good, but notoriously difficult to compare [25]. Although the choice of coding and matching strategies differ significantly between systems, the greatest source of variability is probably the least relevant, i.e., the selection of the particular collection of faces on which to carry out tests and, in particular, the choice of transformation between target and probe over which the system is supposed to perform recognition [26]. The investigations described in this paper are performed using ORL [19] and PICS [20] databases. The face images are cropped to reduce the bulk storage as well as to shorten the training and identification times. The database is divided randomly into different size training and/or test sets. The whole set of images is resampled as 32×32 pixels and converted to 256 gray level images. To reduce the image size, a low pass filter is applied to the image before interpolation using the nearest neighbor interpolation method. This reduces the effect of Moiré patterns and ripple patterns that result from aliasing during resampling.

3.1. **ORL database.** The ORL database [19] was developed at the Olivetti Research Laboratory, Cambridge. The data consists of 400 images acquired from 40 persons, some of which were taken at different times for some of the persons. Figure 1 shows the cropped faces of 40 subjects included in the ORL database. There are variations in facial expression (open/closed eyes and smiling/non-smiling), and facial details (glasses/no glasses). All images were taken against a dark homogenous background with the subjects in an upright frontal position, with tolerance for some tilting and rotation of up to 20 degrees. There is some variation of the scale of up to about 10% (see Figure 2). The images are gray scale with a resolution of 92×112 pixels. The images are size normalized.

3.2. **PICS database.** The PICS (Psychological Image Collection at Stirling) database [20] was developed at the Psychology Department, University of Stirling, UK. It consists of many data sets. The data sets selected for this study are nott-faces-originals and

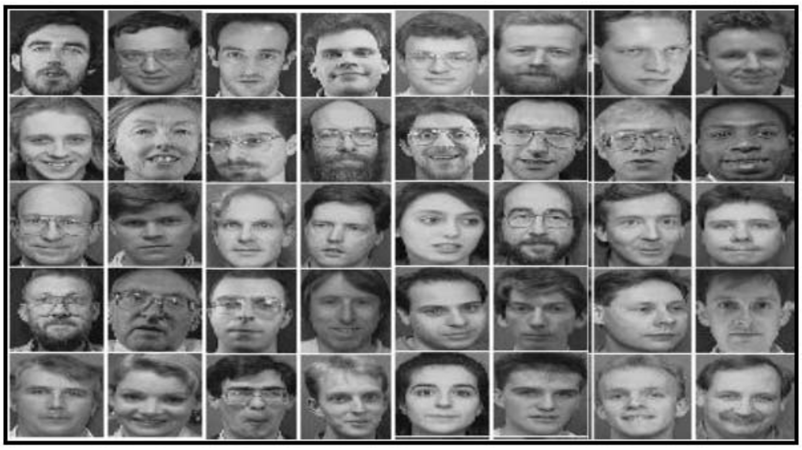

FIGURE 1. Neutral faces of 40 persons for ORL database

FIGURE 2. All face varieties of one person for ORL database

Figure 3. Neutral faces of 70 persons for PICS database

Figure 4. Various face varieties of one person for PICS database

stirling-faces. The database comprises of 490 images of 70 subjects. Neutral cropped images of all subjects are shown in Figure 3. Each face is posed in four expressions with a frontal view, two expressions with a 3/4 view and one frontal view wearing a bathing cap (see Figure 4). The images are available in gif gray level (8-bits) format of variable sizes. The cropped face images are used for the identification of system.

4. **Overview of Approach.** Two of the most important issues for supervised learning can be specified as generalization performance and efficiency. The former addresses the problem of how to develop an adaptive pattern recognition system to achieve optimal performance on samples that are not included in a training set with a finite number of training samples. The latter addresses the complexity of a pattern recognition system in both space and time. The space complexity refers to the size of a system and the time complexity characterizes the computational time needed to develop such systems. To increase the efficiency and the performance of the system, faces are cropped from entire facial images. The developed neural network classifier for face identification is trained on the random training sets. The training samples are individually sized specifically to get the overall performance scenario of the network architecture. The most challenging step in the design of a pattern recognition system is the selection of a suitable base model, which constitute its building blocks. The next step is the features selection and extraction method. Careful selection of a feature extraction method highly simplifies the design of the classifier subsystem. Last step is the performance evaluation for an identification algorithm.

4.1. **Neural classifiers.** In machine-based recognition, a gallery of patterns is first enrolled in the system and coded for subsequent searching. A probe pattern is then obtained and compared with each coded pattern in the gallery and recognition is noted when a suitable match occurs. The challenge of such a system is to perform human identification despite transformations: changes in angle of presentation and lighting, common problems of machine vision and changes of expression and age, which are more special to human faces. The need is, thus, to find appropriate coding for a face which can be derived from one or more images of it and to determine in what way, and how well, two such coding shall match before the faces are declared the same.

Although eigenfaces provide a compact representation of whole faces, they do not provide invariance over changes in scale, head orientation, and lighting conditions [27]. However, in the case of feed-forward neural networks, the face subspace is identified according to the training stage and is application specific. The relationship between the internal representation of knowledge in neural networks and the eigenvector decomposition was studied by several authors [28,29], who have proved that the neural network learning can approximate an eigenvector description of the data presented. Almost, in all cases, carefully designed neural network classifiers [30,31] which are trained on raw image pixels result in superior performance when compared to geometrical feature based methods [32,33].

Thus, all the stages of modern pattern recognition could be performed by a single scheme such as neural networks or genetic algorithms which have the inherent capabilities of noise filtering, data reduction, feature extraction and classification. The advantage of using neural networks and genetic algorithms is that they can extract the most discriminative and representative set of features. The learning vector quantization (LVQ) aims at defining decision surfaces between the competing classes. The decision surfaces obtained by a supervised stochastic learning process of the training data are piecewise-linear hyperplanes that approximate the Bayesian minimum classification error probability.

4.2. **Algorithm.** A generic LVQ neural network consists of three layers. The first layer is the input layer, which consists of as many neurons as the number of input samples of the image to be recognized. The hidden layer size is problem dependent. The number of hidden layer neurons (NH) should be suitable to capture the knowledge of the problem domain. For example, training a neural network to recognize patterns which belong to a certain number of classes (C), at least C hidden layer neurons are required. To capture a large range of input pattern variability, a large number of hidden layer neurons is necessary. However, the problem is how large should it be?

Normally, this number is overestimated by including excess neurons in the network. This means that after convergence, some neurons will be redundant in the sense that they

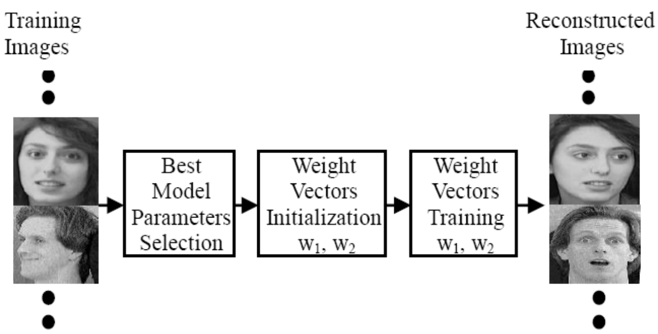

FIGURE 5. LVQ architecture algorithm-flow for face identification

do not evolve significantly and thus do not capture any data structure. The compact LVQ network training algorithm for the identification of faces is illustrated in Figure 5. The algorithm based on the best LVQ parameters [15] is as follows:

1- Select the network parameters:
Input layer size (L) = Image size ($32\times32 = 1024$ neurons)
Number of classes (C) = Number of subjects (S)
Training set size = S (40 or 80 or 120 subjects) $\times$ X (3 or 4 appearances)
Number of hidden layer neurons (H) = S $\times$ Round(X/2)
Learning rate (α) = 0.1
Set up the target vector which specifies the target class of each pattern in the training set
Display update rate = 100
Arrange the input patterns of the training set as one-dimensional columns in an array P
Number of training epochs EP = 4000 for S = 40, 6000 for S = 80, or 8000 for S = 120
2- Initialize an LVQ classifier:
Initialization of the weight matrix for competitive layer w_1 and
Initialization of the weight matrix for linear layer w_2
3- Start training of an LVQ classifier based on selected efficient parameters
4- Test the trained classifier on test sets and compute pccts (percentage of correct classification for test sets)
5- Check if a desired level of identification accuracy is achieved, save the weight vectors or otherwise train the network again
6- Repeat Steps (2)-(5) to get 100 well trained networks
7- Calculate the average of the 100 neural network identification classifiers
8- Exit

The convergence speed of backprop is directly related to the learning rate parameter α. If α is small, the search path will closely approximate the gradient path but convergence will be very slow due to the large number of update steps needed to reach a local minimum. On the other hand, if α is large, convergence initially will be very fast but the algorithm will eventually oscillate and thus not reach a minimum [14]. In general, it is desirable to have large steps when the search point is far away from a minimum, with decreasing step size as the search approaches a minimum. The optimal learning rate for fast convergence of backprop/gradient-descent search is the inverse of the largest eigenvalue of the Hessian matrix of the error function evaluated at the search point. Computing the full Hessian matrix is prohibitively expensive for large networks [14]. Therefore, various experiments are performed to find an optimal value of a ranging from 0.005 to 1.0. The value of α that yields the best performance on the average of 100 runs of the multilayer feedforward network is 0.1 on the face identification task.

Efficiency deals with the complexity of a learning machine in both space and time. The learning time must scale nicely with respect to the size of data sets. Since the size of learning machines determines the memory required for implementation, a learning machine with a compact structure is preferred. Therefore, a challenging problem is how to develop an adaptive learning system with a compact structure to achieve good performance that can be adopted in real time. Reducing the number of hidden-layer neurons of the neural network to the number of subjects multiplied by half the number of training images per subject helps in increasing the efficiency and the performance of the whole system. Thus, both the training time and the test time are minimized. This enables the system to be adopted for real-time applications.

4.3. **Z-based formal specification.** A set of definitions used in the formal model is presented in this section. First of all, we define a matrix of 32×32 in Z notation. Each element in the matrix has a type, value, and property. Type of the element is integer, value is in the range of 0 to 255 gray levels, and property is an abstract set type as defined in the schema *Element* given below. For simple specification, the basic set type is used for the characteristics of an element. In modeling using sets in Z notation, we do not put any restriction upon number of elements of a set and in this way a high level of abstraction is achieved. Further, we do not argue about any systematics procedure for verifying whether an arbitrary element is a member of the set. As a consequent, *Property* is a set over which we cannot define any operation, for example, cardinality to know the number of elements in it. The other operations, for example, subset and complement over *Property* are not defined.

$[Property]$

Element

$type: \mathbb{Z};\ value: 0 \mathrel{..} 255$
$property: Property$

As we know a matrix is a two dimensional array consisting of rows and columns. At first, we define row as a sequence of elements by using the schema *Row* given below. The sequence structure is used because an order of elements is required in defining row of a matrix. In first part of the schema, row is declared which is of type of sequence and invariant is defined in second part called predicate of the schema. In predicate, it is stated that total number of elements in a row must be exactly equal to 32. The cardinality operator is used in defining this invariant which takes a set as input and returns a non-negative number as output.

Row

$row: \mathrm{seq}\ Element$

$\#(\mathrm{dom}\ row) = 32$

A matrix is described by the schema *Object* consisting of two components which are rows and properties. The first one is a sequence of rows which is a type of sequence of *Row*. The second one is a set of properties based on elements of the matrix. Now we have completely defined the matrix as a sequence of elements. The first element of the *rows* is, in fact, first row of the matrix. The second element is second row of matrix and so on. The schema structure is used for composition of the objects because it is very powerful at an abstract level of specification and helps in describing a good specification approach. In the invariants, it is stated that there are 32 columns in the matrix and properties set is a collection of properties of elements of the matrix.

Object

$rows: \mathrm{seq}\ Row$
$properties: \mathbb{F}\ Property$

$\#(\mathrm{dom}\ rows) = 32 \land properties = \{\ i: \mathbb{N};\ r: Row;\ e: Element \mid i \in \mathrm{dom}\ rows$
$\land r \in \mathrm{ran}\ rows \land (i, r) \in rows \land (\forall j: \mathbb{N} \mid j \in \mathrm{dom}\ r.row \land e \in \mathrm{ran}\ r.row \bullet (j, e) \in r.row) \bullet e.property\ \}$

The whole database of face images of all the categories is defined by the schema *Database* as given below. The schema contains *objects/subjects* which is a power set of *Object* and assumed to be a finite set. It is further assumed that total number of face images in the database is equal to *grouplimit* and is defined using axiomatic structure. The axiomatic structure is usually used to define the global constants in Z notation. Before partitioning the *database*, operations required to verify the similarity or distinctness of objects are described below. These operations are important in grouping, testing, and training of subjects.

$$objectlimit : \mathbb{Z};\ grouplimit : \mathbb{Z}$$

Database

$$objects : \mathbb{F}\ Object$$

$$\#\, objects = grouplimit$$

The similarity of two objects is verified using the schema *SameObjects* which consists of two objects to be verified. In the predicate part of the schema, it is stated that objects are same: a) If number of rows in both objects are equal. b) Properties of both objects are same. c) Type, value and property of each element of first object are same to the corresponding element of second object. It is noted that, for a moment, we have used mathematical language of Z which is used to describe various objects. The same language is used to define the relationships between the objects. This relationship is used in terms of constraints after composing the objects in the predicate part of the schema.

SameObjects

$$o1, o2 : Object$$

$$\begin{array}{l}
o1\,.\,rows = o2\,.\,rows \wedge o1\,.\,properties = o2\,.\,properties \\
\forall i : \mathbb{N};\ r1, r2 : Row \mid i \in \mathrm{dom}\ o1\,.\,rows \wedge i \in \mathrm{dom}\ o2\,.\,rows \wedge r1 \in \mathrm{ran}\ o1\,.\,rows \\
\quad \wedge\ r2 \in \mathrm{ran}\ o2\,.\,rows \wedge (i, r1) \in o1\,.\,rows \wedge (i, r2) \in o2\,.\,rows \\
\quad \bullet\ \forall j : \mathbb{N};\ e1, e2 : Element \mid j \in \mathrm{dom}\ r1\,.\,row \wedge j \in \mathrm{dom}\ r2\,.\,row \\
\qquad \wedge\ e1 \in \mathrm{ran}\ r1\,.\,row \wedge e2 \in \mathrm{ran}\ r2\,.\,row \wedge (j, e1) \in r1\,.\,row \\
\qquad \wedge\ (j, e2) \in r2\,.\,row \bullet e1\,.\,type = e2\,.\,type \wedge e1\,.\,value = e2\,.\,value \wedge e1\,.\,property = e2\,.\,property
\end{array}$$

In the schema *DifferentObjects*, given below, dissimilarity of two objects is checked. The schema consists of two objects to be tested. In the predicate part, it is stated that objects are dissimilar: a) If number of rows in both objects are different, b) Properties of objects are not same, c) Values of corresponding elements do not match.

DifferentObjects

$$o1, o2 : Object$$

$$\begin{array}{l}
o1\,.\,rows = o2\,.\,rows \wedge o1\,.\,properties = o2\,.\,properties \\
\exists i : \mathbb{N};\ r1, r2 : Row \mid i \in \mathrm{dom}\ o1\,.\,rows \wedge i \in \mathrm{dom}\ o2\,.\,rows \wedge r1 \in \mathrm{ran}\ o1\,.\,rows \\
\quad \wedge\ r2 \in \mathrm{ran}\ o2\,.\,rows \wedge (i, r1) \in o1\,.\,rows \wedge (i, r2) \in o2\,.\,rows \\
\quad \bullet\ \exists j : \mathbb{N};\ e1, e2 : Element \mid j \in \mathrm{dom}\ r1\,.\,row \wedge j \in \mathrm{dom}\ r2\,.\,row \\
\qquad \wedge\ e1 \in \mathrm{ran}\ r1\,.\,row \wedge e2 \in \mathrm{ran}\ r2\,.\,row \wedge (j, e1) \in r1\,.\,row \\
\qquad \wedge\ (j, e2) \in r2\,.\,row \bullet e1\,.\,type \neq e2\,.\,type \vee e1\,.\,value \neq e2\,.\,value \vee e1\,.\,property \neq e2\,.\,property
\end{array}$$

For a particular problem's experiment, a collection of objects is selected from given database. The selected objects constitute a collection of groups. In each group, there are same number of images. The procedure of partitioning analysis of objects is defined below using the schema *Groups*. In the schema *Groups*, the schema *Database* is used for the creation of the *groups* as in the first part of schema. In the predicate part, the invariants are specified for the well-defined-ness of groups.

$$
\begin{array}{|l}
\textit{Groups} \\
\hline
\textit{Database} \\
\textit{groups}: \mathbb{F}\,(\mathbb{F}\ \textit{Object}) \\
\hline
\forall \textit{group}: \mathbb{F}\ \textit{Object} \mid \textit{group} \in \textit{groups} \bullet \#\textit{group} = \textit{objectlimit} \\
\forall \textit{group}: \mathbb{F}\ \textit{Object} \mid \textit{group} \in \textit{groups} \bullet \textit{group} \subseteq \textit{objects} \wedge \forall \textit{group}: \mathbb{F}\ \textit{Object} \mid \textit{group} \in \textit{groups} \\
\quad \bullet \forall o1, o2: \textit{Object} \mid o1 \in \textit{group} \wedge o2 \in \textit{group} \bullet o1.\textit{rows} = o2.\textit{rows} \wedge o1.\textit{properties} = o2.\textit{properties} \\
\forall \textit{group1}, \textit{group2}: \mathbb{F}\ \textit{Object} \mid \textit{group1} \subseteq \textit{objects} \wedge \textit{group2} \subseteq \textit{objects} \\
\quad \bullet \forall o1, o2: \textit{Object} \mid o1 \in \textit{group1} \wedge o2 \in \textit{group2} \bullet o1.\textit{rows} \neq o2.\textit{rows} \vee o1.\textit{properties} \neq o2.\textit{properties} \\
\forall \textit{group}: \mathbb{F}\ \textit{Object} \mid \textit{group} \in \textit{groups} \bullet \forall o1, o2: \textit{Object} \mid o1 \in \textit{group} \wedge o2 \in \textit{group} \\
\quad \bullet \forall i: \mathbb{N}; r1, r2: \textit{Row} \mid i \in \mathrm{dom}\ o1.\textit{rows} \wedge i \in \mathrm{dom}\ o2.\textit{rows} \wedge r1 \in \mathrm{ran}\ o1.\textit{rows} \\
\quad \wedge r2 \in \mathrm{ran}\ o2.\textit{rows} \wedge (i, r1) \in o1.\textit{rows} \wedge (i, r2) \in o2.\textit{rows} \bullet \forall j: \mathbb{N}; e1, e2: \textit{Element} \mid j \in \mathrm{dom}\ r1.\textit{row} \\
\quad \wedge j \in \mathrm{dom}\ r2.\textit{row} \wedge e1 \in \mathrm{ran}\ r1.\textit{row} \wedge e2 \in \mathrm{ran}\ r2.\textit{row} \wedge (j, e1) \in r1.\textit{row} \wedge (j, e2) \in r2.\textit{row} \bullet e1.\textit{type} \\
\quad = e2.\textit{type} \wedge e1.\textit{value} = e2.\textit{value} \wedge e1.\textit{property} = e2.\textit{property} \\
\forall \textit{group1}, \textit{group2}: \mathbb{F}\ \textit{Object} \mid \textit{group1} \in \textit{groups} \wedge \textit{group2} \in \textit{groups} \\
\quad \bullet \forall o1, o2: \textit{Object} \mid o1 \in \textit{group1} \wedge o2 \in \textit{group2} \bullet \exists i: \mathbb{N}; r1, r2: \textit{Row} \mid i \in \mathrm{dom}\ o1.\textit{rows} \\
\qquad \wedge i \in \mathrm{dom}\ o2.\textit{rows} \wedge r1 \in \mathrm{ran}\ o1.\textit{rows} \wedge r2 \in \mathrm{ran}\ o2.\textit{rows} \wedge (i, r1) \in o1.\textit{rows} \wedge (i, r2) \in o2.\textit{rows} \\
\qquad \bullet \exists j: \mathbb{N}; e1, e2: \textit{Element} \mid j \in \mathrm{dom}\ r1.\textit{row} \wedge j \in \mathrm{dom}\ r2.\textit{row} \\
\qquad \wedge e1 \in \mathrm{ran}\ r1.\textit{row} \wedge e2 \in \mathrm{ran}\ r2.\textit{row} \wedge (j, e1) \in r1.\textit{row} \\
\qquad \wedge (j, e2) \in r2.\textit{row} \bullet e1.\textit{type} = e2.\textit{type} \wedge e1.\textit{value} \neq e2.\textit{value} \vee e1.\textit{property} \neq e2.\textit{property} \\
\hline
\end{array}
$$

Invariants: (1) It is stated that there are same number of elements (face images) in each group which are assumed to be *objectlimit* which is a global constant defined above. (2) Each group of elements is a subset of the original database. (3) All elements (face images) in each group are of same subject or object. (4) Each group is created from the same object and all groups are distinct. (5) It is verified that for any two face images (matrices) in a group, the corresponding elements are same, that is, have the same characteristics. (6) It is also verified that for any two face images there exists at least one element in the first matrix which is different from the corresponding element of the other matrix.

An algorithm for merging two elements is described using the *Merge* schema. There are three components in the schema. The first two, with question mark, are given as input. The third component with exclamation mark is output variable in which a new constructed face image is stored after merging two faces of one subject. In the predicate part, properties of input images are extracted and merged by the union operator, and are stored in output variable.

$$
\begin{array}{|l}
\textit{Merge} \\
\hline
o1?, o2?: \textit{Object}; o!: \textit{Object} \\
\hline
o!.\textit{properties} = \{ i: \mathbb{N}; r: \textit{Row}; e: \textit{Element} \mid i \in \mathrm{dom}\ o1?.\textit{rows} \\
\qquad \wedge r \in \mathrm{ran}\ o1?.\textit{rows} \wedge (i, r) \in o1?.\textit{rows} \wedge (\forall j: \mathbb{N} \mid j \in \mathrm{dom}\ r.\textit{row} \wedge e \in \mathrm{ran}\ r.\textit{row} \bullet (j, e) \in r.\textit{row}) \bullet e. \\
\quad \textit{property} \} \cup \{ i: \mathbb{N}; r: \textit{Row}; e: \textit{Element} \mid i \in \mathrm{dom}\ o2?.\textit{rows} \\
\qquad \wedge r \in \mathrm{ran}\ o2?.\textit{rows} \wedge (i, r) \in o2?.\textit{rows} \\
\qquad \wedge (\forall j: \mathbb{N} \mid j \in \mathrm{dom}\ r.\textit{row} \wedge e \in \mathrm{ran}\ r.\textit{row} \bullet (j, e) \in r.\textit{row}) \bullet e.\textit{property} \} \\
\hline
\end{array}
$$

The training algorithm is described using the schema *Training* given below. There are two components in the schema. The first one is *Groups* schema used for verification of trained objects. The second is the collection of trained objects, *trained!*, which are required to be verified. In the predicate part of the schema, it is stated that for any object *o* in the trained set, if there exists a group in the database which contains two same objects

$o1$ and $o2$, then these objects are merged and stored in o using *Merge* schema discussed above.

Training

$$\begin{array}{l}
Groups;\ trained!: \mathbb{F}\ Object \\
\hline
\forall to: Object \mid to \in trained! \bullet \exists group: \mathbb{F}\ Object \mid group \in groups \\
\quad \bullet \forall o1, o2: Object \mid o1 \in group \wedge o2 \in group \quad \bullet \forall i: \mathbb{N};\ r1, r2: Row \mid i \in \mathrm{dom}\ o1\ .\ rows \\
\quad \wedge\ i \in \mathrm{dom}\ o2\ .\ rows \wedge r1 \in \mathrm{ran}\ o1\ .\ rows \wedge r2 \in \mathrm{ran}\ o2\ .\ rows \wedge (i, r1) \in o1\ .\ rows \wedge (i, r2) \in o2\ .\ rows \\
\quad \bullet \forall j: \mathbb{N};\ e1, e2: Element \mid j \in \mathrm{dom}\ r1\ .\ row \wedge j \in \mathrm{dom}\ r2\ .\ row \\
\quad \wedge\ e1 \in \mathrm{ran}\ r1\ .\ row \wedge e2 \in \mathrm{ran}\ r2\ .\ row \wedge (j, e1) \in r1\ .\ row \wedge (j, e2) \in r2\ .\ row \\
\quad \bullet\ e1\ .\ type = e2\ .\ type \wedge e1\ .\ value = e2\ .\ value \wedge e1\ .\ property = e2\ .\ property \\
\quad \Rightarrow to\ .\ properties = o1\ .\ properties \cup o2\ .\ properties
\end{array}$$

The testing algorithm is described using the schema *Testing*. There are two components in the schema consisting of *Training* and $o!$. The first is *Training* schema used for testing of an object. The second one is the object $o!$ itself to be tested. In the predicate part, it is stated that the testing is successful if there exists an object o in the trained objects such that the type, value and properties of $o!$ and o are exactly the same.

Testing

$$\begin{array}{l}
Training;\ o?: Object \\
\hline
\exists to: Object \mid to \in trained! \bullet \forall i: \mathbb{N};\ r1, r2: Row \mid i \in \mathrm{dom}\ o?\ .\ rows \wedge i \in \mathrm{dom}\ to\ .\ rows \\
\quad \wedge\ r1 \in \mathrm{ran}\ o?\ .\ rows \wedge r2 \in \mathrm{ran}\ to\ .\ rows \wedge (i, r1) \in o?\ .\ rows \wedge (i, r2) \in to\ .\ rows \\
\quad \bullet \forall j: \mathbb{N};\ e1, e2: Element \mid j \in \mathrm{dom}\ r1\ .\ row \wedge j \in \mathrm{dom}\ r2\ .\ row \\
\quad \wedge\ e1 \in \mathrm{ran}\ r1\ .\ row \wedge e2 \in \mathrm{ran}\ r2\ .\ row \wedge (j, e1) \in r1\ .\ row \\
\quad \wedge\ (j, e2) \in r2\ .\ row \bullet e1\ .\ type = e2\ .\ type \wedge e1\ .\ value = e2\ .\ value \wedge e1\ .\ property = e2\ .\ property
\end{array}$$

5. **Empirical Results and Discussion.**

A number of experiments have been performed to explore the possibility of the best parameters selection for an invariant adaptive face identification network architecture. Tables 1 and 2 show various results of the identification networks trained with ORL and PICS databases. The identification rate of the network is 100% for a random test set of 10 subjects with 60 faces for the ORL database (see Table 1), that is, 6 face varieties like, facial expression (open/closed eyes, smiling/non-smiling), facial details (glasses/no glasses) per subject. The system is trained with a training set of 40 images of the same subjects. The efficiency is evaluated on both the time and the space scale. Setting NH equal to S×round(X/2) has reduced the network memory requirements for the internal representation of target faces. In addition, it enhances the processing speed and both training time of the network and identification time of the face image are reduced. Table 2 depicts the network results for the PICS database. The identification rate is 100% for a random test set of 10 subjects with 30 faces. It may be noticed that few face images per subject are used for this database as the number of available varieties is less. This implies that the proposed architecture is quite capable of recognizing more target faces efficiently and is independent of the source and the size of training data. Some researchers attempt to verify their developed recognition systems by using only small training sets of uniform size and draw conclusions based on this about the system's efficiency and performance. We have divided databases into different size random training and test sets to get a clear scenario of the architecture's performance.

Performance addresses the generalization capability of a learning machine on randomly chosen samples that are not included in a training set. The performance of a supervised

TABLE 1. pccts of face identification for ORL with parameters $\alpha = 0.1$, IS $= 32 \times 32$ pixels

S	EP	I	Tr	Ts	NH	peak pccts	Ave. pccts
10	1000	100	40	60	20	100.0	100.0
20	2500	200	80	120	40	99.16	98.51
30	4000	300	120	180	60	98.89	98.05
40	5500	400	160	240	80	96.67	95.50

S = subjects, EP = epochs, I = images (total),
Tr = training set, Ts = test set,
NH = number of hidden neurons, IS = image size,
pccts = percentage of correct classification of test set

TABLE 2. pccts of face identification for PICS with parameters $\alpha = 0.1$, IS $= 32 \times 32$ pixels

S	EP	I	Tr	Ts	NH	peak pccts	Ave. pccts
10	1000	60	30	30	20	100.0	100.0
20	2500	120	60	60	40	98.33	98.10
30	4000	180	90	90	60	97.78	97.11
40	5500	240	120	120	80	95.83	95.00

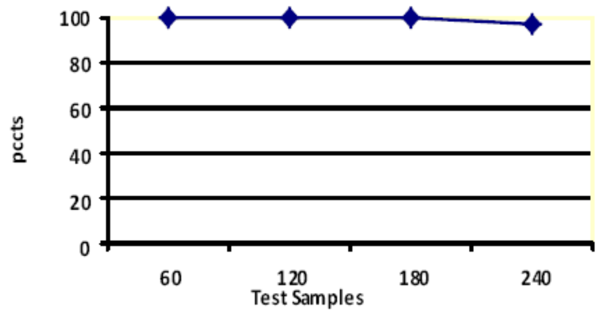

FIGURE 6. Performance: pccts vs. test samples for ORL database

learning system is also characterized by its generalization error, which measures the distance between the output function of a trained model and an underlying target function. If a neural network is too large, it may over fit a particular training set and thereby fail to maintain good generalization error. A small neural network, however, may be sufficient to approximate an optimal solution [34].

The generalization superiority of the neural net can be attributed to the bounded and smooth nature of the hidden-unit responses. Once particular approximation function or network architecture is decided on, generalization can be improved if the number of free parameters in the net is optimized [14]. Thus, eliminating the redundant hidden-layer neurons of the network architecture not only makes the system more efficient but also increases the system performance (see Figure 6).

The experiments are repeated on larger data sets including 400 images (40 subjects $\times$ 10 faces) for ORL and 240 images (40 subjects $\times$ 6 faces) for the PICS database. The purpose of this experiment is to show that the neural network architecture is more efficient for larger databases than for smaller ones. This advantage represents major progress toward solving the scalability problem in pattern recognition tasks.

6. **Conclusions.** The need of high-speed computing machines, database and networking technologies, and sophisticated image processing methodologies has increased the topical significance of automatic identification systems which usually deploy optimal or adaptive classifiers to achieve reasonable classification results. In modern pattern recognition systems, all the stages of pattern recognition could be performed by a single scheme such as neural networks or genetic algorithms, which has the inherent capabilities of noise filtering, data reduction, feature extraction and classification. Mostly in existing work, testing is carried through simulation but unfortunately this approach is lacking in verifying the correctness of such systems. Further, the number of simulations increases exponentially to provide a required level of confidence due to their complexity. When a modification to the system is needed, regression testing will be required to perform which suggests that the complete set of simulations must be re-conducted. The most important is that the performance promises obtained through simulations are not absolute even if it does provide a conditional proof that the system is correct under certain assumptions. Validation and verification are processes that help to ensure that models and simulations are correct and reliable. That is why formal methods are supported to simulation in this research and it provided us to verify the system properties in a rigorous way. In this paper, performance evaluation synthesizing Z-based formal specification and empirical analysis is presented for an adaptive small-sized and low resolution face-image identification algorithm. Z is a model oriented specification language based on set theory and first order predicate logic and is used for verification of an invariant adaptive neural network based face identification algorithm. The Z/Eves tool is used because it is a powerful one for specification analysis and provides various exploration techniques to prove the properties of a system.

We have achieved both objectives: i) automatic identification of human faces using an adaptive invariant neural network architecture that is trained on small-sized and low-resolution images, and ii) performance evaluation for an identification algorithm based on formal specification. The architecture has shown its feasibility using a collection of 890 face images of 110 subjects from two standard databases; ORL [20] and PICS [21]. Empirical results yield a recognition accuracy rate of 96.7% on 40 subjects and 240 face images for ORL. The advantage of using neural networks and genetic algorithms is that they can extract the most discriminative and representative set of features. They offer fine-grained parallelism and exhibit fault tolerance.

Acknowledgment. This work is supported by Scientific Research Deanship, King Faisal University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K. Sakai, M. Nagao and T. Kanade, Computer analysis and classification of photographs of human faces, *The 1st USA-Japan Computer Conf.*, vol.2, no.7, pp.55-62, 1972.

[2] A. L. Yuille, D. S. Cohen and P. W. Halliman, Feature extraction from faces using deformable templates, *Proc. of IEEE Conf. Comp Vision and Patt. Recog.*, pp.104-109, 1989.

[3] C. C. Chibelushi, J. S. D. Mason and F. Deravi, Feature-level data fusion for bimodal person recognition, *Proc. of the 6th IEEE Int. Conf. Image Proc. App.*, pp.399-403, 1997.

[4] C. C. Chibelushi, F. Deravi and J. S. Mason, Audio-visual person recognition: An evaluation of data fusion strategies, *Proc. of the 2nd Eur. Conf. Security Detection*, pp.26-30, 1997.

[5] U. Meier, W. Hurst and P. Duchnowski, Adaptive bimodal sensor fusion for automatic speech reading, *Proc. of IEEE Int. Conf. Acoustics Speech Sig.*, vol.2, pp.833-836, 2008.

[6] R. Raghavendra, R. Ashok and K. G. Hemantha, Subjective performance of texture based algorithm for face verification: The role of databases, *IEEE Proc. of the 6th Indian Conf. Comp. Vision Graphics & Image*, pp.421-428, 2008.

[7] J. Martins and J. P. Hubaux, Evaluating performance from formal specification, *IEEE Proc. of MASCOTS'96*, pp.285-290, 1996.
[8] J. Bowen and V. Stavridou, Safety critical systems, formal methods and standards, *Software Engineering Journal*, vol.8, no.4, pp.189-209, 1993.
[9] J. P. Bowen and M. G. Hinchey, Seven more myths of formal methods, *The 5th Eur. Soft. Eng. Conf.*, Barcelona, Spain, pp.25-28, 1995.
[10] A. S. Samaria, *Face Recognition Using Hidden Markov Models*, Ph.D. Thesis, Trinity College University of Cambridge, UK, 1994.
[11] *The Psychological Image Collection at Stirling (PICS)*, http://pics.psych.stir.ac.uk/.
[12] D. A. Stuart, M. Brockmeyer, A. K. Mok and F. Jahanian, Simulation-verification: Biting at the state explosion problem, *IEEE Trans. Soft. Eng.*, vol.27, no.7, pp.599-617, 2001.
[13] J. L. Bentley, *Programming Pearls*, Addison-Wesley, 1989.
[14] J. M. Spivey, *The Z Notation: A Reference Manual*, Prentice-Hall Upper, Saddle River, NJ, USA, 1989.
[15] I. Meisels and M. Saaltink, *The Z/Eves Reference Manual, TR-97-5493-03 ORA*, Canada, 1997.
[16] M. J. Er, W. Chen and S. Wu, High-speed face recognition based on discrete cosine transform and RBF neural networks, *IEEE Trans. Neural Networks*, vol.16, no.3, pp.679-691, 2005.
[17] G. Guo and C. R. Dyer, Learning from examples in the small sample case: Face expression recognition, *IEEE Trans. Systems Man and Cybernetics B: Cybernetics*, vol.35, no.3, pp.477-488, 2005.
[18] H. Zhao, P. C. Yuen and J. T. K. Wok, A Novel incremental principal component analysis and its application for face recognition, *IEEE Trans. Systems Man and Cybernetics B: Cybernetics*, vol.36, no.4, pp.873-886, 2006.
[19] Y. Liu and Y. Chen, Face recognition using total margin-based adaptive fuzzy support vector machines, *IEEE Trans. Neural Networks*, vol.18, no.1, pp.178-192, 2007.
[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Analysis and Mach. Intel.*, vol.31, no.2, pp.210-227, 2009.
[21] C. Hsieh, S. Lai and Y. Chen, An optical flow-based approach to robust face recognition under expression variations, *IEEE Trans. Image Processing*, vol.19, no.1, pp.233-240, 2010.
[22] M. Nishiyama, A. Hadid, H. Takeshima, J. Shotton, T. Kozakaya and O. Yamaguchi, Facial deblur inference using subspace analysis for recognition of blurred faces, *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol.33, no.4, pp.838-845, 2011.
[23] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, England, 1995.
[24] G. A. Khuwaja, LVQ base models for recognition of human faces, *Int. J. Comput. Appl. Tech.*, vol.16, no.4, pp.181-193, 2002.
[25] G. Robertson and I. Craw, Testing face recognition systems, *Imag. Vis. Computing*, vol.12, pp.609-614, 1994.
[26] I. Craw, N. Costen, T. Kato and S. Akamatsu, How should we represent faces for automatic recognition, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol.21, no.8, pp.725-736, 1999.
[27] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, Eigenfaces ve. Fisherfaces: Recognition using class specific linear projection, *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol.19, no.7, pp.711-720, 1997.
[28] A. Valentin and H. Abdi, Can a linear autoassociator recognize faces from new orientations, *J. of Opt. Society of America*, vol.13, no.4, pp.717-724, 1996.
[29] M. Gross, *Visual Computing*, Springer-Verlag, NY, 1995.
[30] S. Lawrence, G. C. Lee, A. C. Tsoi and A. B. Andrew, Face recognition: A convolutional neural network approach, *IEEE Trans. on Neural Networks*, vol.8, no.1, pp.98-113, 1997.
[31] S. H. Lin, S. Y. Kung and L. J. Lin, Face recognition/detection by probabilistic decision-based neural network, *IEEE Trans. on Neural Network*, vol.8, no.1, pp.114-132, 1997.
[32] J. Ben-Arie and D. A. Nandy, Volumetric/iconic frequency domain representation for objects with application for pose invariant face recognition, *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol.20, no.5, pp.449-457, 1998.
[33] L. Kin-Man and Y. Hong, An analytic-to-holistic approach for face recognition based on a single frontal view, *IEEE Trans. Pattern Analysis and Mach. Intell.*, vol.20, no.7, pp.673-686, 1998.
[34] G. A. Khuwaja, Merging face and finger images for human identification, *Patt. Anal. and App. Journal*, vol.8, no.1-2, pp.188-198, 2005.