A new Hierarchical Pattern Recognition method using Mirroring Neural Networks

Dasika Ratna Deepthi
Department of Computer Science,
Aurora's Engineering College, Bhongir,
Nalgonda Dist., A.P., India

K. Eswaran
Department of Computer Science,
Srinidhi Institute of Science and Technology,
Yamnampet, Ghatkesar, Hyderabad, India.

ABSTRACT
In this paper, we develop a hierarchical classifier (an inverted tree-like structure) consisting of an organized set of "blocks" each of which is actually a module that performs a feature extraction and an associated classification. We build each of such blocks by coupling a Mirroring Neural Network (MNN) with a clustering (algorithm) wherein the functions of the MNN are automatic data reduction and feature extraction which precedes an unsupervised classification. We then device an algorithm which we name as a "Tandem Algorithm" for the self-supervised learning of the MNN and an ensuing process of unsupervised pattern classification so that an ensemble of samples presented to the hierarchical classifier is classified and then sub-classified automatically. This tandem process is a two step process (feature extraction/data reduction and classification), implemented at each block (module) and can be extended level by level in the hierarchical architecture. The proposed procedure is practically demonstrated using 2 example cases where in a collage of images consisting of faces, flowers and furniture are classified and sub classified automatically.

We expect that this kind of architecture will be very useful in the development of efficient and powerful self-learning machines in the future.

Categories and Subject Descriptors
I.5.1 **[Pattern Recognition]:** Models – Neural nets

General Terms
Algorithms

Keywords
Hierarchical Pattern Recognition, classifier, feature extraction, Mirroring Neural Networks, unsupervised classification, Tandem Algorithm, self-supervised learning.

1. INTRODUCTION
There have been various ways in which the fields of artificial intelligence and machine learning have been furthered: starting with experimentation [1], abstraction [2], [3] and the study of locomotion [4]. Many techniques have been developed to learn patterns [5], [6] as well as to reduce large dimensional data [7] & [8] so that relevant information can be used for classification of patterns [9] & [10]. Investigators have tackled, to varying degrees of success, pattern recognition problems like face detection [11], gender classification [12], human expression recognition [13], object learning [14] & [15], unsupervised learning of new tasks [16] and also have studied complex neuronal properties of higher cortical areas [17], naming but a few. However, most of the above techniques did not require automatic feature extraction as a pre-processing step to pattern classification. In our approach, we developed a self-learning machine which performs feature extraction and pattern learning simultaneously to recognize/classify the patterns in unsupervised mode. This automatic feature extraction step, prior to unsupervised classification fulfills one more additional crucial requirement called dimensional reduction. In this paper we show that these two machine steps, namely, automatic feature extraction and clustering can be performed by the modules (blocks) of suitably designed hierarchical classifiers. This modular approach is a hierarchically extendable procedure, involving a tree-like architecture, which performs a level-by-level unsupervised classification of the given input. Each module of the proposed architecture is actually an abstraction of an entity which executes two procedures: the "Mirroring Neural Network" (MNN) algorithm coupled with "Forgy's clustering algorithm". The MNN performs automatic data reduction and feature extraction and Forgy's clustering does the subsequent step called unsupervised classification (of the extracted features of the MNN); these two steps are performed in tandem - hence our term "Tandem Algorithm".

We find it necessary to discuss a few points about the MNN before moving on to the details of the Tandem Algorithm. An MNN is nothing but a neural network with converging-diverging architecture which is trained to produce an output which equals its input as closely as possible (i.e. mirror the input at its output layer). And this training process proceeds through repeated presentations of all the input samples and it stops when the MNN could mirror at least above 95% of its input samples. Then the MNN said to be successfully trained with the given input samples. Now, the best possible extracted features of the inputs are automatically obtained at the MNN's least dimensional hidden layer and these features are used for unsupervised input classification by a clustering algorithm. See Figure 1 for illustration of an MNN architecture wherein input given to it is 'X' of dimension n which is reduced to 'Y' of dimension m (m is much less than n). Since Y is capable of mirroring X at the out put, Y contains as much information as X, even though it has a lesser dimension, the components of Y can then be thought of as features that contains the patterns in X, hence, the Y can be used

for classification. More details on MNN architecture can be referred from [18].

The Tandem Algorithm which we devised in this paper for pattern recognition tasks using a hierarchical tree-like architecture (depicted in Figure 2) may be considered as an extension to Kolmogrov's theorem [19] in providing a practical method for unsupervised classification. It may be noted that each block in the hierarchical architecture is trained through the implementation of a single common algorithm (tandem algorithm). This tandem process is done (at each node) in two steps. The 1st step being the data reduction and feature extraction by an MNN and the 2nd step is the classification of the extracted features (outputs of the MNN) using a clustering algorithm. The MNN at the first level trains itself to extract the features through repeated presentations of the data samples, after which the extracted features of the data are sent to the clustering procedure for classification. The modules in the second level again undergo this tandem process of feature extraction and classification (/sub-classification). This is how a single common algorithm is implemented throughout the hierarchy (at each module), resulting a level-by-level unsupervised classification. In section 3, we show that our method actually works: we apply our classifier on a collage of images of flowers, faces and furniture, this collection is automatically classified and sub classified. The details of the mirroring neural network and the mathematical base which justify the techniques used in the development of the hierarchical classifier (discussed in this paper) can be obtained from [18] and [20].

Before, proceeding to the presentation of actual computer simulation, it is perhaps appropriate to write a few lines on the ideas that motivated this paper. It is presently well known that the neural architecture in the human neocortex is hierarchical [21], [22], [23] & [24] and constituted by neurons at different levels and information is exchanged between these levels via these neurons [25], [26], [27] & [28] when initiated by the receipt of data coming in from sensory receptors in the eyes, ears, etc. The organization of the various regions within each level of the neo-cortical system, are not completely understood, but there is much evidence that regions of neurons in one level are connected with regions of neurons in another level thus forming many tree like structures [24] & [29] (also see [30]). Various intelligent tasks, for example "image recognition", are performed by numerous neurons firing and sending signals back and forth these levels [31]. Many researchers working in the field of artificial intelligence have sought to imitate the human brain in their attempt to build learning machines [32] & [33] and have employed a tree like architecture at different levels for performing recognition tasks [34] and have used statistical techniques to extract features from data [35]. As described above, our attempt here is to demonstrate that a hierarchical classifier which addresses the tasks of feature extraction (/data reduction) and recognition can be constructed and such architecture can perform intelligent recognition tasks in an unsupervised manner.

The plan of the paper is as follows: In the next section we show how to build pattern classifiers which possess the ability to automatically extract features, have a tree-like architecture (see Figures 1 to 3) and can be used to develop the proposed architecture for unsupervised pattern learning and classification (including the proposed tandem algorithm). In section 3, we report the results of such a classifier when applied to two specific unsupervised pattern recognition problems wherein real images of faces, flowers and furniture are automatically classified and sub-classified in an unsupervised manner (see Figure 4). Section 4, we discuss the future possibilities of this kind of architecture.

2. UNSUPERVISED HIERARCHICAL PATTERN CLASSIFIER

This section describes the architecture of a self-learning engine and the next section, we report its application to two specific examples, wherein a set of input images are automatically classified and then sub-classified.

Our intent is to build a learning engine which has the following characteristics: It is (i) hierarchical (ii) modular (iii) unsupervised and (iv) runs on a single common algorithm (MNN associated with clustering). The advantage of developing a recognition system with these 4 characteristics is that the learning method does not depend on the problem size and the learning network can be simply extended as the recognition task becomes more complex. It has been surmised by investigators that the architecture of human neo-cortex does, loosely speaking, possess the above 4 characteristics (except that instead of (iv) there is some kind of analog classification process (procedure) performed by sets of neurons, which seemingly behave in a similar manner). We are also reinforced by the conviction, since our architecture imitates the neural architecture (though admittedly in a crude manner), it is reasonable to expect that we would meet with greater successes as we make improvements in our techniques and as we deal with problems of larger size using increasingly powerful computers. In fact, it is this prospect that has been the prime motive force behind our work.

We will now develop the tandem algorithm and actually implement it by writing a computer program by which such a learning engine can be used to classify the patterns by itself and report the results. The technique used for the development of this algorithm is based upon the application of the two procedures (i) mirroring for automatic feature extraction and (ii) unsupervised classification, in a tandem manner as described by the following algorithm, at each module (block) of the hierarchy, level-by-level. (In our computer program we have used it on a two level hierarchy).

Continuing the discussion of the Tandem Algorithm, consider Figure 2 which is a pictorial representation of the hierarchical architecture, the details of each block or node is shown in Figure 3 and the structure of a MNN in Figure 1. The tandem Algorithm proceeds block (node) by block (node) at each level starting from the 1st level (see Figure 2).

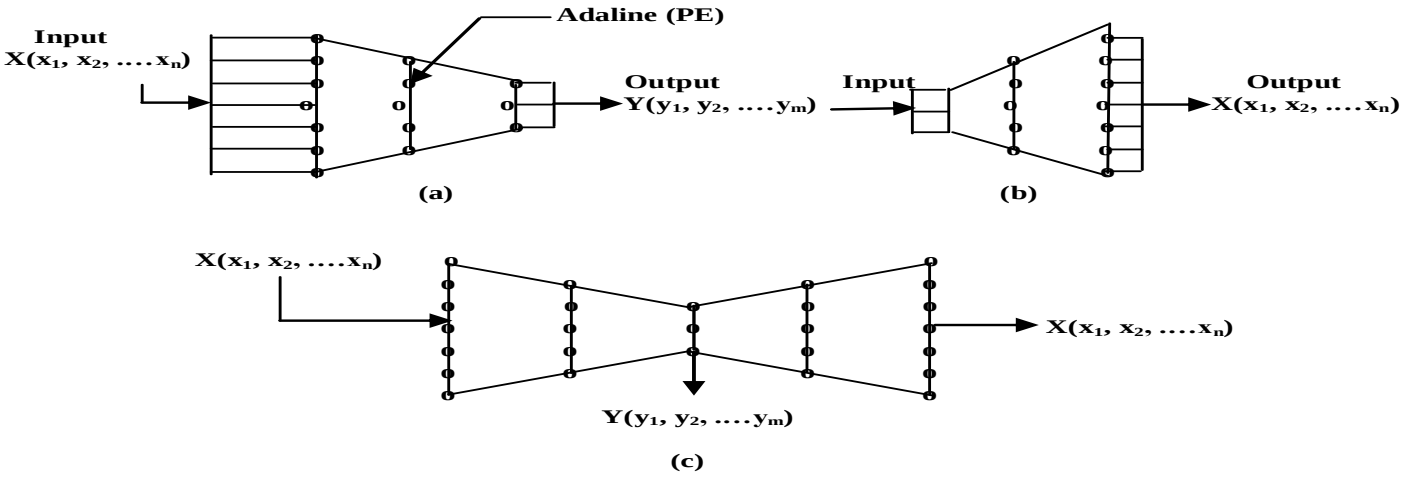

Figure 1. (a) Convergning NN (b) Diverging NN (c) Mirroring Neural Network (combining (a) and (b))

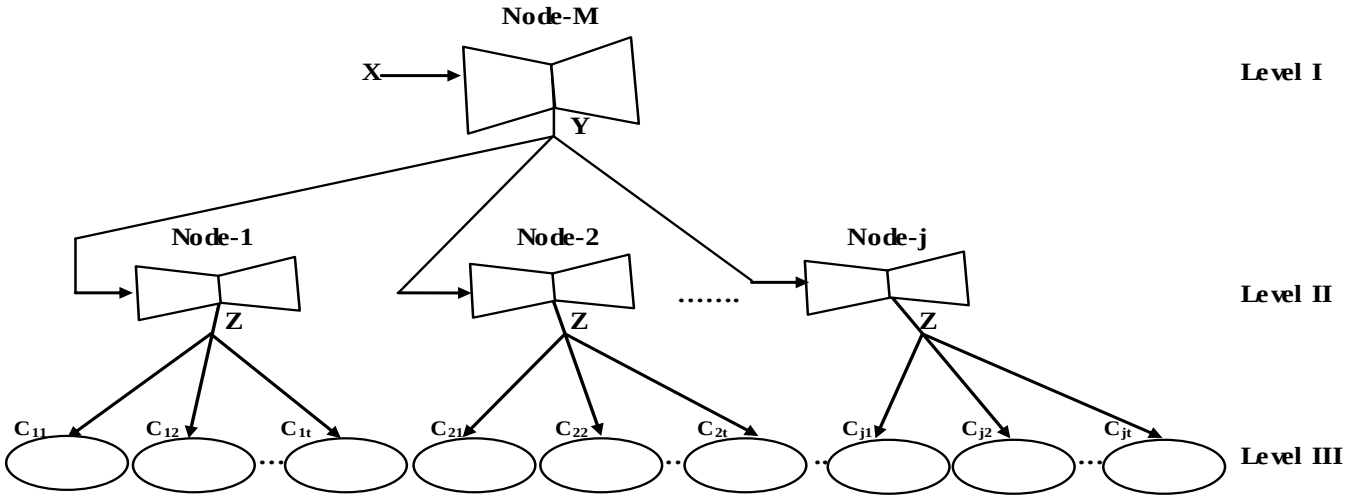

Figure 2. Organized collection of Nodes (blocks) containing MNN's and their corresponding Forgy's algorithm – Forming a treelike hierarchical structure

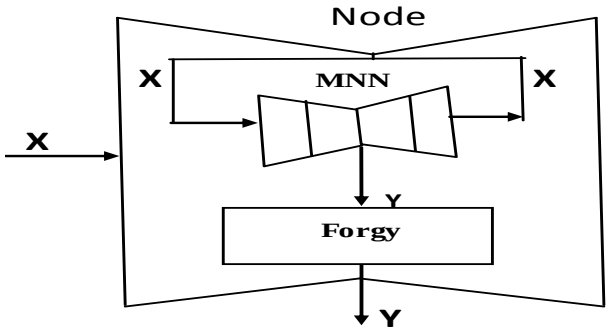

Figure 3. A Node (block) of the Hierarchical Classifier constructed with MNN and Forgy's clustering

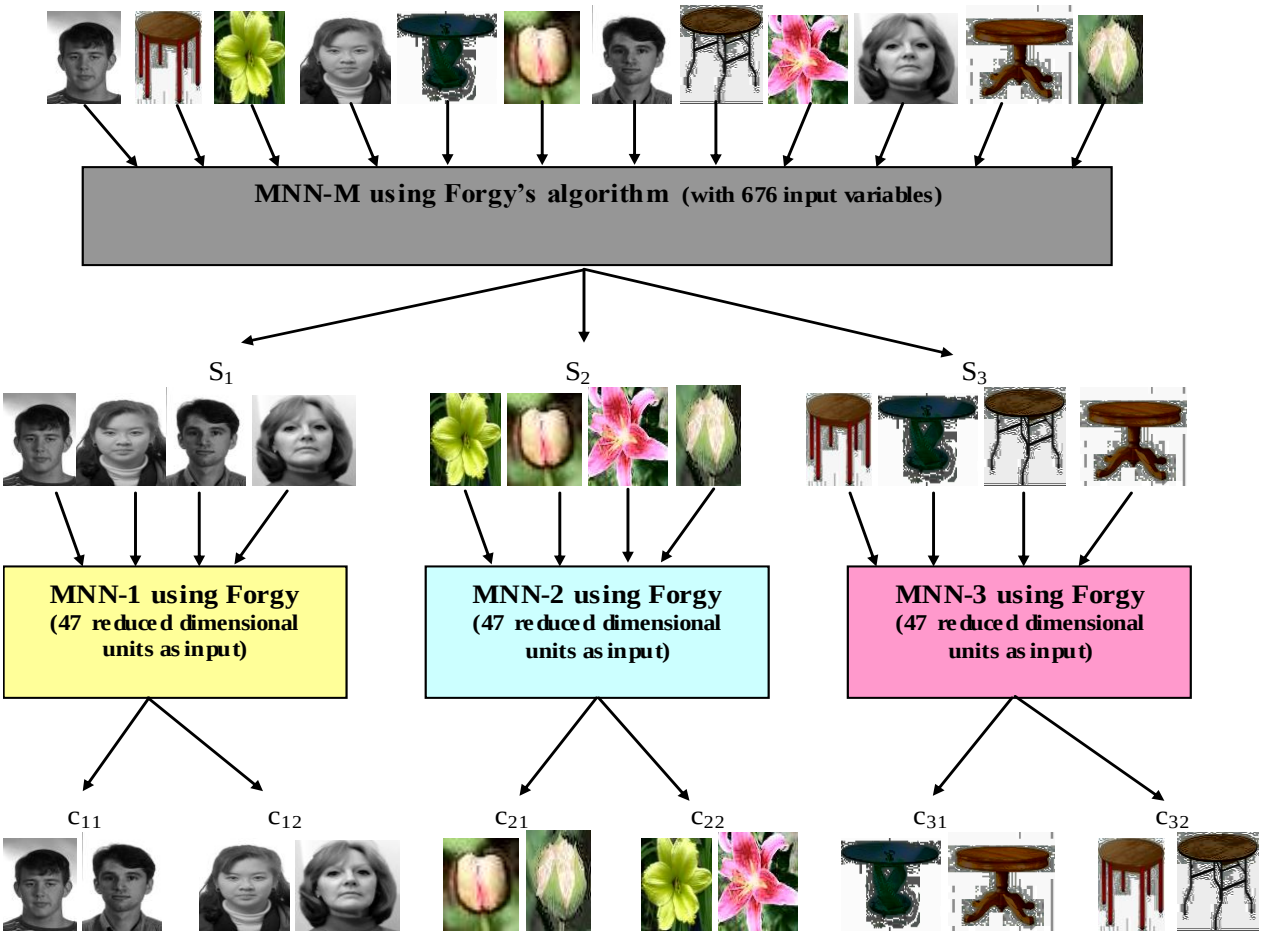

Figure 4 Pictorial representation of Hierarchical classifier implemented using Example 2 images. (S_1 (face), S_2 (flower), S_3 (table): classification at level I based on 47 reduced dimensional vectors of the input image; c_{11} (male face), c_{12} (female face), c_{21} (flower bud), c_{22} (open flower), c_{31} (centrally supported table), c_{32} (four-legged table): sub-classification at level II based on 30 reduced dimensional vectors of the image).

The Tandem Algorithm for a hierarchical classifier:

1. Train the MNN of the topmost block, i.e. Node-M (of the hierarchy) with an ensemble of samples such that the inputs given to the MNN are mirrored at its output with minimum loss of information (for which a suitable threshold is fixed). And mark the topmost node as the "present node". This is an iterative step and stops when the output of Figure 1c, almost equals the input, that is able to reconstruct the input.

2. After due training of the MNN of the present node (i.e., the MNN could accurately reconstruct above 95% of its inputs within the specified threshold limit), the collection of outputs of the MNN's least dimensional hidden layer (the extracted number of features is equal to the dimension of Y of the MNN see Figure 1c) is considered for classifying the input of the present node.

3. The features extracted in step 2 are given as "input data set" to the Forgy's clustering algorithm (subroutine) of the present node for unsupervised classification, explained in step 4.

4. The steps involved in clustering procedure are:

   a. Select initial seed points (as many in number as the no. of classes the inputs to be divided into) from "input data set".

   b. For all the data samples in the input dataset repeat this step b.

      (i) Calculate distance between each sample in the input data set and the each of the seed points representing a cluster.

      (ii) Place the input data sample into the group associated with the seed point which is closest (least of the distances in step 4 b (i)).

   c. When all the samples are clustered, the centroids of each of the clusters are considered as new seed points.

   d. Repeat step 4 b, 4 c as long as the data sets leave one cluster to join another in step 4 b (ii).

5. To proceed further for sub-classification, repeat step 6 for all the nodes in the next below level of the hierarchy.

6. Mark one of nodes as the "present node" and train the MNN of the present node with the samples (extracted features of the immediate above level) belong to a particular cluster of step 4 such that the samples given to the MNN are mirrored at its

output with minimum loss of information (for which a threshold is fixed by trail and error). Repeat steps 2, 3 and 4.

7. Repeat steps 5 and 6 till there is no enough data present in the samples to be further sub-classified (at the immediate below level).

In this tandem algorithm, the feature extraction (concurrent with data reduction) is through steps 1, 2 and 3 and the automatic data classification (based on the reduced units of data) is by step 4. This tandem process of data reduction and classification is extended to next lower levels of the hierarchy for sub-classifying the ensemble through steps 5 and 6 till the stated condition is met in step 7.

More details on the MNN architecture and MNN's training through self-learning are given in [18] & [20].

We now, illustrate this concept of hierarchical architecture for unsupervised classification using Figure 2. If we assume, for the purpose of illustration, that there are only 4 categories of images; say faces, flowers, furniture and trees (j = 4), then at its broadest level, the MNN-M at Node-M is trained with these 4 categories of images. On successful training, MNN-M can reduce the amount of its input data; and based on the reduced units of data, Node-M categorizes the pattern into one of the classes using Forgy's algorithm. The reduced units (which represent the input data) of the pattern from the present node (Node-M) are fed to one of the next level (Level II) nodes. (Alternatively, the input vector could be fed to the appropriate MNN in next level (Level II), instead of the reduced vector, in cases where too large an amount of data reduction done at the present level (Level I), is expected to have loss of information required for the finer classification at Level II). Selection of a node (module) from next level depends upon the classification of the input pattern at the present level. For example, Node-1 is selected if Node-M classifies the input as a face, else Node-2 is selected if Node-M classifies the same input as a flower and so on for Node-3 (furniture) or Node-4 (tree). Then, the respective node (module) at Level II reduces its input and does a sub-classification (we denote it as Level II classification) based only on its reduced units (at Level II). The gender classification which distinguishes a male face from a female face is a typical Level II classification by Node-1. In the pictorial representation, Level II classification contains 't' subcategories in each of j categories. Assuming that there are some more lower levels (identical to Level I and/or Level II) containing the nodes to further classify the patterns, so, for instance, the reduced units at Level II are given as input to one of the appropriate modules at Level III for more detailed classification which, an example case, sub-categorizes 'k' different persons in male/female face group. This tandem procedure of (i) mirroring followed by (ii) classification, performed at each level, can be extended to other lower levels, say, level IV, V and so on. That is how; the proposed architecture does level-by-level unsupervised pattern learning and recognition.

As explained earlier, the hierarchical architecture implements a common algorithm for data reduction and extracted feature classification at its each node. And as the data reduction precedes the classification, the accuracy of classification is dependent on the efficiency of the data reduction algorithm. So there is a need to evaluate the performance of the MNN's data reduction. The fact that the MNN dimensional reduction technique is an efficient method to reduce the irrelevant parts of the data was amply demonstrated over extensive trials (details are in [18] & [20]). It is because of this that we used the MNN (along with clustering algorithm) as a data reduction and feature extraction tool for the hierarchical pattern classification. For our demonstration use the Forgy's algorithm for clustering the reduced units (of the input, at each module), wherein the number of clusters for the classification/sub-classification is provided by the user. Instead, without prejudice to the generality of our technique, one could use a more sophisticated clustering algorithm wherein the number of classes (clusters) is determined by the algorithm. We leave this work as a part of future enhancement which would then result in a completely automated unsupervised classification algorithm.

3. DEMONSTRATION AND RESULTS

We now show by explicit construction that a hierarchical architecture can actually be built and used for classification and sub-classification of images, giving two example cases.

Example 1: Here we took a collection of 300 images approximately half of them are face images (See databases Feret [36], Manchester [37], Jaffe [38] in references) and the other half are images of various tables. We build a two level classifier constructed out of MNNs (associated with Forgy's clustering); which at the first level automatically classifies the 300 images of the training set into two classes one of them would be a "face class" and the other the "table class". The automatic procedure which does this is as follows: A 4 layer MNN (676-60-47-676) consisting an input layer of 676 inputs representing a 26 X 26 image, with the 60 processing elements in the first layer and 47 and 676 processing elements in the other two layers is used to train the MNN to mirror the input data on to itself. The training is done automatically and stops when the output vector of 676 dimensions closely matches the corresponding input vector for each image, at this point, the MNN can said to be satisfactorily mirror all the 300 input images.

Then the output of the layer with the least number of processing elements (in this case 47) is taken as a reduced feature vector for the corresponding input image. We would have a set of 300 vectors (each of 47 dimensions) representing the input data of 300 images. This set of 300 vectors (of reduced dimensions) is then classified into two classes by using Forgy's Algorithm (see [39] & [40]). The actual classification varies somewhat on the choice of the initial seed points which are randomly chosen. The program chooses two distinct initial random seed points and uses Forgy's algorithm to cluster the reduced vectors of the input images. This clustering is done over many iterations till convergence and the classes are then frozen; after this the data is clustered a second time (starting from the second set of seed points) again using Forgy's algorithm till another set of two classes are obtained. After this the average of the resulting two nearest sets of cluster centroids is considered as the new cluster centroid, based on which the reduced feature vectors are once again classified to obtain two distinct classes, these classes are then treated as the final two classes (if everything works out well one of them would be the face class and the other the table class).

After this first level classification, the program proceeds to the next level for sub-classifying the two classes identified at level I. The procedure of reduction and classification at this Level II, is similar to that carried out at Level I, except that now two MNNs have to be trained, one receiving inputs form the Face class and the other from the Table class . These MNNs at Level II use the

architecture (47-37-30-47). After the two MNNs are suitably trained to mirror their respective inputs, to an acceptable accuracy, the program proceeds to classify the inputs into sub categories for each of the MNNs separately. Of course, this time the feature vector (reduced vectors) has 30 dimensions. Once again, Forgy's Algorithm is used, following a similar procedure as described above for level I, except that this time the classification is done on the reduced vectors of the MNN at Node-1 which would render the sub categories male face and female face, followed by a classification of the reduced vectors of the MNN at Node-2 obtaining the subcategories centrally supported table and four legged table. Because the MNNs are initiated with random weights (chosen initially), and again by choosing random seed points while executing the Forgys Algorithm, it is our intention to demonstrate that the classification is not overly dependent on these random choices. So, we ran the program over and over again each time starting ab initio, all the results are reported herein. We have taken 10 trials meaning, 10 different training and classification sessions at level I followed by level II. On an average, at level I, the accuracy of classification is 97.5%. That is the level I node could automatically classify the faces from the tables with an error of only 2.5%. Since the classification is unsupervised, with each trial (initiated by random choices of seed points), the network classifies the input into one of two classes and it really does not label the first class as "face class" and the second as "tables class", so it could happen that the sets will be interchanged. This does not pose problems because then at the second level the labels will be again interchanged. After this the training and classification at level II starts, the 47 dimension feature vectors belong to the 1st class (say face class) is fed to the MNN at Node-1 and those that belong to the second class (table class) are fed to the MNN at Node-2. After satisfactory training of the two MNNs, the classification into subclasses is performed by their respective Forgy's clustering. The average classification accuracy of both the nodes at level II is 95% i.e., based on the 30 feature vectors of the input image is grouped into a male face or a female face or a centrally supported table or a four legged table with a cumulative error of 5%, because the level II classification is consequent to the categorization of the input into a face or a table with an error of 2.5%. Actually, this is not too bad at all, since the whole exercise is unsupervised and the errors made in the 1st level classification remain undiscovered and are actually uncorrected by the classifier which indiscriminately feeds all the data into the second level as inputs. In spite of all this, the average error for the overall classification (for the entire architecture) is within 5%.

Example 2: In this example we have taken the MNN architectures for level I and level II to be the same as that of the Example 1 and data set consists of 360 images for training with an equal no. of faces, tables and flowers and the results show that even for this example the images are successfully classified and sub-classified.

The error at level I, it is 6% and an average error of the three nodes at level II (for subcategorizing a "face" as "male" or "female", a "table" as "centrally supported" or "four-legged" and a "flower" as "flower bud" or "open flower") is 12%. The results are averaged over 10 trails similar to the first example. See the sample illustration for the Example 2 in Figure 4. The summary of the results for Example 1 and Example 2 are given in Table 1 and Table 2 respectively. The various parameters used in the MNN training and classification are given in Table 3.

The brute force (obvious procedure) of training the MNN at each node of the hierarchical classifier by using a Newton-Raphston is beyond the capability of the PCs available with us and was not tried. Instead, we adopted an approximate procedure and trained the MNNs by using the Back Propagation algorithm ([41] & [42]) which actually tries to determine the best MNN by changing the weights at each presentation of an image; ideally a "best MNN" should be obtained for the entire ensemble of input images (or reduced units of images) at each MNN of a node, which again would involve a Newton-Raphston and was avoided. The techniques used and reported here were very efficient in terms of time and space taken for execution and they were all performed on a PC.

Table 1. Results of the hierarchical classifier for Example 1

Input type	Dimension of the input	Dimension of the reduced units	No. of samples for training	No. of samples for testing	No. of categories	Success rate (averaged over 10 trails) of clustering on reduced units	
						Average of Training & testing	Average of Testing samples
Image	**676** (26 X 26)	**47**	**300**	**100**	**2** (face & table)	**97.1% (Efficiency of Level I Node)**	**96.2% (Efficiency of Level I Node)**
Reduced units of image	**47**	**30**	**≈ 150** (for each category)	**≈ 50** (for each category)	**2** (sub-categories for each category)	**93.6%** (average of the two level II Nodes)	**89.6%** (average of the two level II Nodes)

Table 2. Results of the hierarchical classifier for Example 2

Input type	Dimension of the input	Dimension of the reduced units	No. of samples for training	No. of samples for testing	No. of categories	Success rate (averaged over 10 trails) of clustering on reduced units	
						Average of Training & testing	Average of Testing samples
Image	**676** (26 X 26)	**47**	**360**	**150**	**3** (face, table & flower)	**93.4%** (Efficiency of the Level I Node)	**92.1%** (Efficiency of the Level I Node)
Reduced units of image	**47**	**30**	**≈ 120** (for each category)	**≈ 50** (for each category)	**2** (sub-categories for each category)	**86.3%** (Average efficiency of the three level II Nodes)	**81.3%** (Average efficiency of the three level II Nodes)

Table 3. Various parameters used for Example #1and Example #2

Type of MNN architecture	Distance between input and output	Seed points for Forgy (Example #1)	Seed points for Forgy (Example #2)	Learning rate parameter	Weights& bias terms
Level I MNN (676-60-47-676)	0.8	Threshold of 0.8, between the random seed points	Threshold of 1.0, between the random seed points	0.025	-0.25 to +0.25 (random selection)
Level II MNNs (47-37-30-47)	0.8	Threshold of 0.8, between the random seed points	Threshold of 0.8, between any two random seed points	0.01	-0.25 to +0.25 (random selection)

4. CONCLUSIONS AND FUTURE WORK

In this paper we have developed a crucial Pattern Recognition architecture, called a hierarchical classifier which can be built by using "Mirroring Neural Networks" and an associated clustering algorithm. We have also specifically built two self-supervising classifiers as examples. These classifiers have the characteristics of being hierarchical, modular, unsupervised and they run on a single common algorithm and therefore, they mimic (admittedly in a crude manner), the collective behavior of neurons in the neo-cortex. It is expected that they can be expanded to analyze much more complex data, such "super classifiers" could employ many structures, (each being of the type shown in Figure 2), working in parallel.

In our experimentation, (within the available resources) we have found that it is not possible to have too many classes at the first level (Figure 2), i.e. j cannot be too large a value (at best j = 4). Therefore, for large problems involving many classes, we need to have a network of "structures" (each being of the type shown in Figure 2 but with j limited to 2, 3 or 4) working in parallel, each structure trained to recognize its own set of classes (eg. face classes, alphabet classes etc.). Thus a binary or tertiary "super-tree" with each "node" itself being a structure of type shown in Figure 2, can be envisaged for the construction of a "super classifier".

It is expected that the techniques that we have developed and presented in this paper will be implemented by many future researchers for building advanced and highly sophisticated pattern classifiers. Further it is also hoped that these procedures will also be used for building models for associative memories [43] where, say a voice signal (eg. "Mary": a spoken word) can be associated with a picture (image of Mary). These developments could, in the near future, lead to very versatile machine learning systems which can possibly ape the human brain in at least its elemental functions.

5. REFERENCES

[1] Buchanan, B.G. 1994. The role of experimentation in A.I. In Phil. Trans. R. Soc. A, 349, 153-166.

[2] Zucker, J.D. 2003. A grounded theory of abstraction in artificial intelligence. In Phil. Trans. R. Soc. B, 358, 193-1309.

[3] Holte, R.C. & Choueiry, B.Y. 2003. Abstraction & reformulation in A.I. In Phil. Trans. Roy. Soc.B, 358, 1197-1204.

[4] Cruze, H., Durr, V. & Schmitz, J. 2007. Insect walking is based on a deetralized architecture revealing a simple and robust controller. In Phil. Tran. R. Soc. A, 365, 221-250.

[5] Freund, Y. & Schapire, R.E. 1996. Experiments with a new boosting algorithm. In Proc. 13th International Conference on Machine Learning, pp. 148-156.

[6] Viola. P, & Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1, I-511-I-518 (2001).

[7] Hinton, G.E. & Salakhutdinov, R.R 2006. Reducing the Dimensionality of Data with Neural Networks. In Science 313, 504-507.

[8] Law, H.C. 2006. Clustering, Dimensionality Reduction and Side Information, Ph. D. Thesis, Michigan State University.

[9] Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In Proc. 10th European Conference on Machine Learning, 137-142.

[10] Craven, M., DiPasquo, D., Freitag, D., McCallum, A.K. & Mitchell, T.M. 2000. Learning to construct knowledge bases from the World Wide Web. In Artificial Intelligence 118, 69-113.

[11] Garcia, C. & Delakis, M. 2004. Convolutional face finder: A neural architecture for fast and robust face detection. In IEEE Trans. Pattern Anal. Mach. Intell. 26, 1408-1423.

[12] Phung, S.M. & Bouzerdoum, A. 2007. A Pyramidal Neural Network For Visual Pattern Recognition. In IEEE Transactions on Neural Networks 18, 329-343.

[13] Rosenblum, M., Yacoob, Y. & Davis, L.S. 1996. Human expression recognition from motion using a radial basis function network architecture. In IEEE Trans. Neural Networks 7, 1121-1138.

[14] Baldi, P. & Harnik, K. 1989. Neural networks and principal component analysis:learning from examples without local minima. In Neural Networks 2, 53-58.

[15] DeMers, D. & Cottrell, G. 1993. Non-linear dimensionality reduction. In Advances in Neural Information Processing Systems 5, Morgan Kaufmann, 580-587.

[16] Hopfield, J.J & Brody, C.D. 2004. Learning rules and network repair in spike-timing-based computation networks. In Proc. Natl. Acad. Sci. U. S. A. 101, 337-342.

[17] Lau, B., Stanley, G.B. & Dan, Y. 2002. Computational subunits of visual cortical neurons revealed by artificial neural networks. In Proc. Nat. Acad. Sci. USA 99, 8974-79.

[18] Deepthi, D.R., Kuchibhotla, S. & Eswaran, K. 2007. Dimensionality reduction and reconstruction using mirroring neural networks and object recognition based on reduced dimension characteristic vector. In IEEE International Conference on Advances in Computer Vision and Information Technology (IEEE, ACVIT-07), 348-353.

[19] Kolmogorov,A.N. 1957. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. In Doklady Akademia Nauk SSSR 114(5), 953-956.

[20] Deepthi, D.R. 2009. Automatic pattern recognition for applications in image processing and robotics, Ph. D. Thesis, Osmania University, Hyderabad, India.

[21] Creutzfeldt, D.O. 1977. Generality of the functional structure of the Neocortex. In Naturwissenschaften 64, 507-517.

[22] Mountcastle, B.V. 1978. An organizing principle for cerebral function: The unit model and the distributed system. In The Mindful Brain, Edelman,G.M, and V.B. Mountcastle,V.B. Eds., Cambridge, Mass.: MIT Press.

[23] Felleman, D.J. & Van Essen, D.C. 1991. Distributed hierarchical processing in the primate cerebral cortex. In Cerebral Cortex 1, 1-47.

[24] Rao, R.P. & Ballard, D.H. 1999. Predictive coding in the visual cortex: A functional interpretation of some extra-classical-receptive-field effects. In Nature Neuroscience 2, 79-87.

[25] Sherman, S.M. & Guillery, R.W. 2002. The role of the thalamus in the ow of information to the cortex. In Phil. Trans. Roy. Soc. London 357, 1695-708.

[26] Kaiser, M. 2007. Brain architecture: A design for natural computation. In Phil. Trans. Roy. Soc. A, 365, 3033-3045.

[27] Buzsaki, G., Geisler, C., Henze, D.A. & Wang, X.J. 2004. Interneuron diversity series: Circuit complexity and axon wiring economy of cortical interneurons. In Trends Neurosci.27, 186-193.

[28] Johnsen, J. D., Santhakumar, V., Morgan, R. J., Huerta, R., Tsimring, L., and Soltesz, I., 2007. Topological determinants of epileptogenesis in large-scale structural and functional models of the dentate gyrus derived from experimental data. In J. Neuro-physiol. 97, 1566-1587.

[29] Van Essen, D.C., Anderson,C.H., & Felleman, D.J. 1992. Information processing in the primate visual system: an integrated systems perspective. In Science 255, 419-423.

[30] Hawkins, J. 2005. On intelligence, Owl Books, Henry Holt & Co., New York, pp. 110-125.

[31] Bell, B.G. 1994. Levels & loops: the future of artificial intelligence & neuroscience. In Phil. Trans. R. Soc. B, 354, 2013-2030.

[32] Hawkins, J. & George, D. 2007. Hierarchical Temporal Memory, Concepts, Theory, and Terminology. In Numenta (Numenta Inc), pp. 1-20 www.numenta.com.

[33] George, D. 2008 How the brain might work: A hierarchical and temporal model for learning and recognition. Ph. D. Thesis, Stanford University.

[34] Herrero, J., Valencia, A. & Dopazo, J. 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. In Bioinformatics 17, 126-136.

[35] Alex, L.P.T., Jacek, M.Z., Lai-Ping, W. & Xu, J. 2007. The hierarchical fast Learning artificial neural network

(HieFLANN) An autonomous platform for hierarchical neural network construction. In IEEE Trans. Neural Networks 18, 1645-1657.

[36] FERET database:www.frvt.org/FERET/.

[37] MANCHESTER database: www.ecse.rpi.edu/ cvrl/database/.

[38] JAFFE database:www.kasrl.org/jaffe.html

[39] Gose, E., Johnsonbaugh, R. & Jost, S. 2000. Pattern Recognition and Image Analysis, Prentice Hall of India, New Delhi, pp 211-213.

[40] Deepthi, D.R., Krishna, G.R.A. & Eswaran, K. 2007. Automatic pattern classification by unsupervised learning using dimensionality reduction of data with mirroring neural networks.

In IEEE International Conference on Advances in Computer Vision and Information Technology (IEEE, ACVIT-07), 354-360.

[41] Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986. Learning Representations by back-propagating Errors. In Nature 323, 533-536

[42] Widrow, B., & Lehr, M.A. 1990. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. In Proceedings of the IEEE 78 (9)

[43] Deepthi, D.R. & Eswaran, K. 2009. Pattern recognition and memory mapping using mirroring neural networks. In IEEE International Conference on Emerging Trends in Computing (IEEE, ICETiC 2009),India, 317-321.