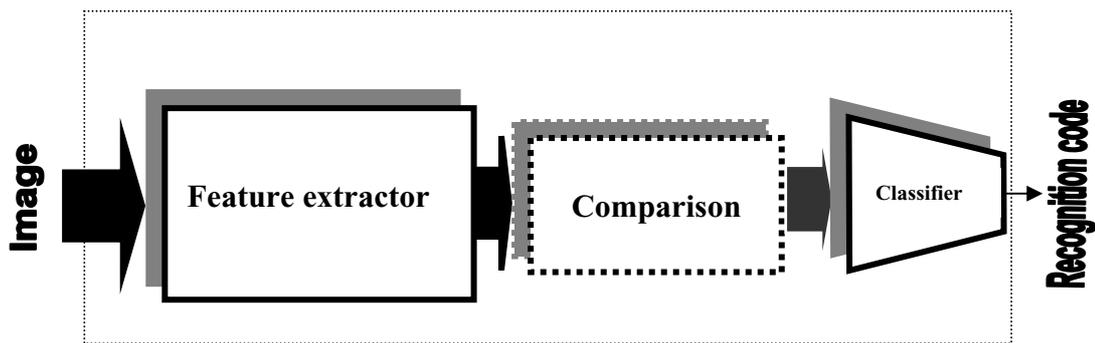


Introduction

An important process in recognition system is the selection of the smaller set of appropriate features from a much bigger set. Selection of 'good' features is critical to performance of recognition and classification [1].

Most of image recognition systems are based on feature extraction at earlier stages, then the features are classified to generate a recognition code for each class to be recognized(see fig. (1)). The features could be extracted by any ways, such as Wavelet [3], Gabor filter [4], PCA (Principle Component Analysis) [1], or Neural networks [2].



Fig(1) Image Recognition Steps

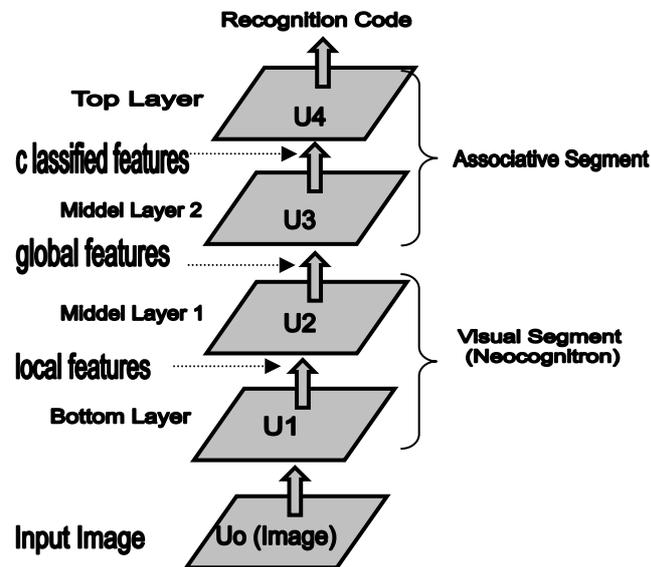
Artificial neural network has the ability to learn classification tasks from examples. However the use of ANNs in the domain of object recognition depends crucially on the 'quality' of feature vector extracted from the image to be classified [8]. Feature extraction should deliver a low dimensional and 'easy to classify' vector, otherwise the network required for the classification task becomes too large and needs too much computational resources, and even worse, requires a huge set of training examples.

2 Image Recognition Neural Network System

The image recognition system described here consists of a hierarchy of several layers of artificial neurons, arranged in planes to form layers. The system consists of layers divided into top, middle, and bottom layers. There can be one or more middle layers. The architecture of the system is divided into two segments: visual and associative segment as shown in figure (2). The visual segment operates on an input image and generate features, which are then processed by associative segment. The visual segment may contain one or two layers (each layer consists of a pair of S and C-sublayers referred to a simple and complex layers respectively).

An image is divided by the visual segment into sub-images[9]. A set of neurons is assigned to each sub-images for classification and in order to produce appropriate codes depending on local features. The extraction of local features is based on the similarity among sub-images. The visual segment trained usually by using unsupervised training algorithm. The training will be implemented

sequentially layer by layer. The output of each layer can be considered as the input to the next layer.



Fig(2) Image Recognition Neural Network System

The number of layers in the visual segment depends on the complexity of the input images. The bottom layer extracts local features from the image as we mentioned above. The role of the next middle layer(s) of the visual segment is to aggregate local features extracted by the bottom layer and generate semi-global features.

The associative segment combines the global features and associates them with correct recognition codes. This segment consists of one or two layers of neurons. The output of the visual segment constitutes input to the associative segment.

The main role of the associative segment is to relate the features generated by the visual segment to the desired recognition code. The associative layers usually feedforward fully connections. The associative segment is trained by supervised training algorithm.

Briefly, we can say that the neocognitron in the designed system is considered as a visual segment that extracts the global features of the input image. Then an associative segment is added to the network to make the global features associated with the recognition code that is labeled by the target of the feedforward top layer.

If two images belonging to the same category of training set have different global features that result from the output of the C-sublayer of the neocognitron, then the associative segment will associate these two different global features to the same recognition code. This can be considered as an advantage of the associative segment.

In this paper an approach to build a good visual segment will be presented. This task depends on which training algorithm is to be used to build such effective one.

3 Neural Network Feature Extraction

The objective of the feature extraction module is to identify the spectral classes present in the image and to define the set of the correspondent samples to be used in the classification phase afterwards.

One of the approaches for feature extraction is the use of competitive learning resulting in data clustering. In this approach, 8x8 pixel windows taken from the original images are used as training pattern. For the competitive layer, four learning algorithms are used; competitive learning [5], modified competitive learning [6], self organizing feature map [7], and modified bias competitive learning which is developed in this paper.

The effectiveness of extracted features differs from type to type. Thus, the aim is to find most effective features. That's mean, each resulted feature should be robust for the whole training images and no feature should produce as a lateral one.

3.1 Competitive Learning (CL)

One of the limitations of competitive networks is that some neurons may not always get allocated. In other words, some neuron weight vectors may start out far from any training patterns, because they are initialize randomly. These neuron never win the competition, no matter how long the training is continued. The result is that their weights do not get to learn and they never win. These unfortunate neurons referred to as 'dead neurons', never perform a useful function[5].

3.2 Modified Competitive Learning (MCL)

The neuron numbers and weights in this algorithm are not initialized randomly, instead of that, neurons are generated during training and the initial values of their weights vectors are equalized to input pattern vectors. MCL can adaptively create a new neuron for an incoming pattern if it determined (by similarity measure) to be sufficiently different from the existing clusters. Notice that one must carefully select the threshold value for the distance between cluster so that the clusters will not be too large to incur high reproduction error, or too small to lose generalization accuracy.

In this algorithm, the dead neurons are guaranteed to disappear, but it was found that some neurons could be produced as a lateral ones, that means the features represented by the weight vectors of such neurons don't represent the most active features which we want to generate. Consequently, they are related to few input patterns. Such neurons are called 'semi-dead neurons' [6].

3.3 Modified Bias Competitive Learning (MBCL)

Each neuron result at the end of learning by this algorithm is active and effective, because of using the biases. Adding bias for each neuron used in the last will prevent it to be a dead neuron or semi-dead neuron. Biases are used to give neurons that only win the competition rarely (if ever) an advantage over neurons which win often. A positive bias, added to the negative distance, makes a distant neuron more likely to win. To do this job, a running average of neuron outputs is kept. It is equivalent to the percentages of time each output is 1. This average is used to update the biases so that the biases of frequently active neurons will get smaller, and the biases of infrequently active neurons will get larger.

The result is that biases of neurons which haven't responded very frequently will increase versus biases of neurons that have responded frequently. As the biases of infrequently active neurons increase, the input space which that neuron responds increases. As that input space increases the infrequently active neuron responds and moves toward more input vectors. Eventually the neuron will response to an equal number of vectors as other neurons. This has two good effects:

First, if a neuron never win a competition because its weights are far from any of the input vectors, its bias will eventually get large enough so that it will be able to win. When this happens, it will move toward some group of input vectors. Once the neurons weights have moved into a group of input vectors and the neuron is winning consistently its bias will decrease to 0. Thus the problem of dead neurons is resolved.

The second advantage of biases is that they force each neuron to classify roughly the same percentage of input vectors. Thus, if a region of the input space is associated with a larger number of input vectors than another region, the more densely filed region will attract more neurons and be classified into smaller subsections. Thus the problem of semi-dead neurons is resolved.

3.4 Self Organizing Feature Map (SOFM)

Self organizing feature map learn to classify input vectors according to how they are grouped in input space. They differ from competitive layers in that neighboring neurons in the self organizing maps learn both the distribution (as competitive layers), and the topology of the input vectors they are trained on [5][7].

Here a self-organizing feature map network identifies a winning neuron using the same procedure as employed by a competitive learning algorithm. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen rule.

Feature maps allocate more neuron to recognize parts of the input space where many input vectors occur and allocate fewer neurons to parts of the input space where few input occur. Each neuron results at the end of learning by this algorithm is active despite of not using the biases here.

4 Experimental Results

The image recognition system presented here was applied to recognize gray level face's images. The size of each image was 64x64 pixels with pixels having 256 gray levels. Each plane of the first simple system's sub layer contains neurons which are extracting a particular local feature like an oriented bar or edge. The size of the local input window of each neuron in each plane in this sub layer has been chosen to be 8x8 pixels.

The SOM used to create the first simple sub layer was composed of 8x8 neurons, while when the other competitive algorithms were used; the competitive layer was composed of 36 neurons. After the SOM and CL's training, each neuron was used to create a particular plane by weight sharing (each neuron represents a particular feature).

A comparison among feature extracted by three types of competitive learning (CL, MCL, and MBCL) are shown on table (1) and in figs (3), (4), (5). Two measures for evaluating the efficiency of each type were given: the running average of each neuron outputs during the training was kept and shown in table (1).

Form of feature extracted by each type at the end of training were built as shown in figures (3), (4), and (5).

One can see that each neuron in MBCL responded to an equal number of vectors as other neurons, the winning average of each neuron trained by this algorithm were equal (0.0277 see table 1). There is neither dead neuron nor semi-dead neuron (see fig (3)).

The winning averages of all neurons trained by MCL are not equal. Lateral features could be seen in fig (4). These features represent the sime-dead neurons. Dead neuron could be seen in Fig (5). The weight of such neurons did not get to learn and they never win (the winning average are zeros).

The Kohonen SOFM provides advantages over other competitive learning techniques, because it provides a graphical organization of pattern relationship (see fig (6)). SOFM achieved significantly better results than MBCL when it is used to train the simple sub layer of the designed recognition system. For example, the recognition rate achieved by using SOFM is better than that was achieved by MBCL when 84 images were used for training and 216 images were used for testing (see table (2)). Thus, SOFM gave good generalization for small training sample size (see also table (2)).

Table(1) Neurons winning average at the end of competitive learning

index	CL _{aver}	MCL _{aver}	MBCL _{ave}	index	CL _{aver}	MCL _{aver}	MBCL _{aver}
1	0	.0457	.0277	18	0	.0209	.0277
2	0	.0268	.0277	19	0	.0118	.0278
3	0	.0109	.0278	20	.0936	.0298	.0277
4	0	.0116	.0277	21	.0398	.0284	.0277
5	.0881	.0177	.0277	22	.0488	.0518	.0277
6	.0540	.0334	.0278	23	.0836	.0448	.0277
7	.0398	.0485	.0277	24	.0436	.0571	.0277
8	.0507	.0341	.0277	25	0	.0118	.0277
9	0	.0122	.0278	26	.0166	.0421	.0277
10	.0507	.0653	.0277	27	0	.0302	.0277
11	.0085	.0225	.0278	28	.0132	.0602	.0277
12	.0952	.0108	.0277	29	0	.0162	.0278
13	0	.0063	.0277	30	.0613	.0125	.0277
14	0	.0258	.0277	31	0	.0075	.0278
15	.0302	.0129	.0277	32	0	.0087	.0277
16	.0429	.0998	.0277	33	.0684	.0122	.0277
17	0	.0568	.0277	34	.0195	.0008	.0278

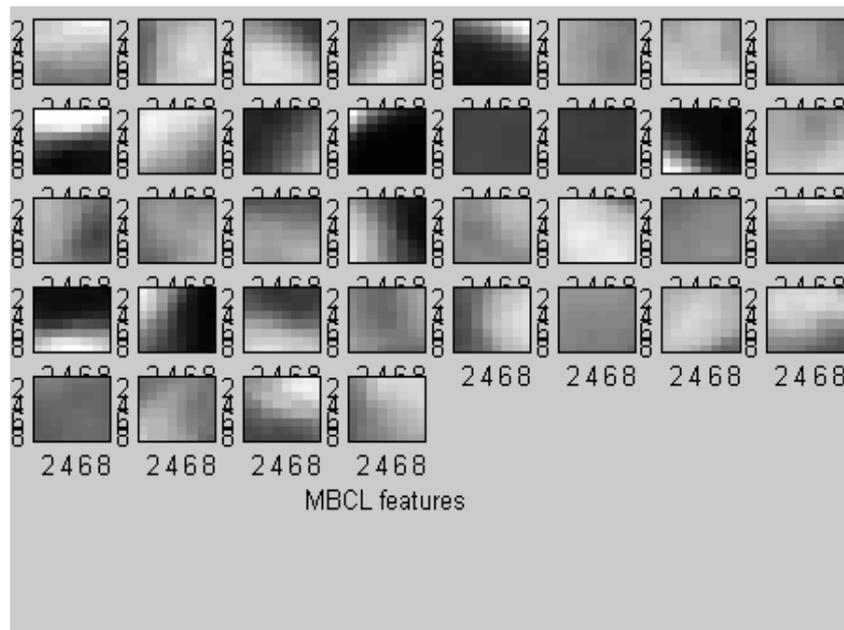
Dead neuron : 
 Semi-dead neuron: 
 index : neuron index
 CL_{aver} : CL neuron winning average
 MCL_{aver} : MCL neuron winning average
 MBCL_{aver} : MBCL neuron winning average

Table(2) Recognition rate achieved by two feature extraction algorithms

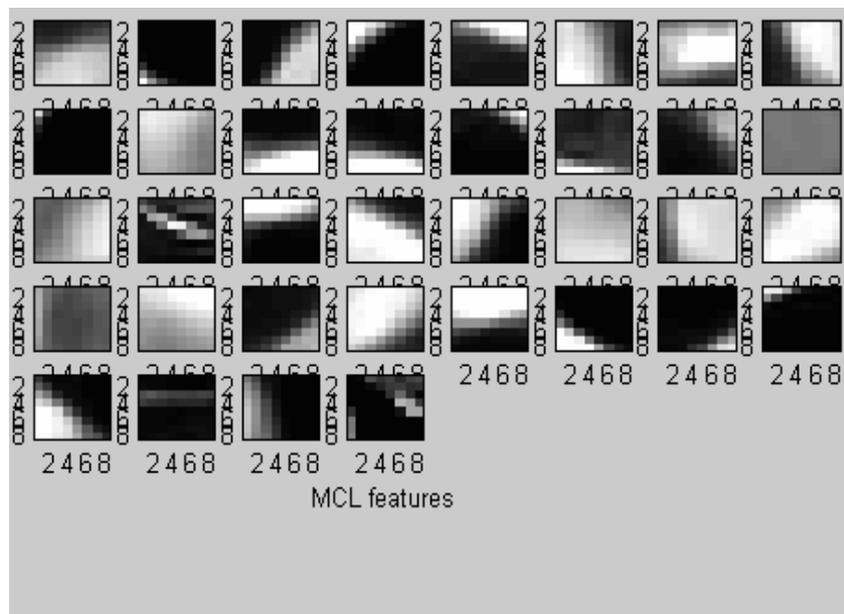
	SOFM	MBCL
C	K1/K2=64/36	K1/K2=42/40
84	92%	86.4%
112	----	91%

C : Number of training images
 K1* : Number of the first layer planes
 K2 : Number of the second layer planes

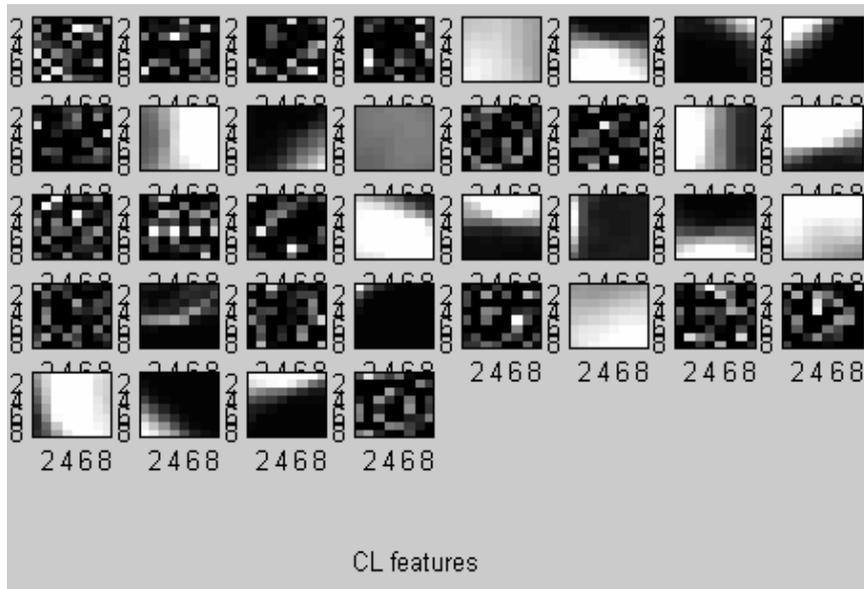
*The number of each layer planes were selected experimentally so that the best recognition rate was achieved.



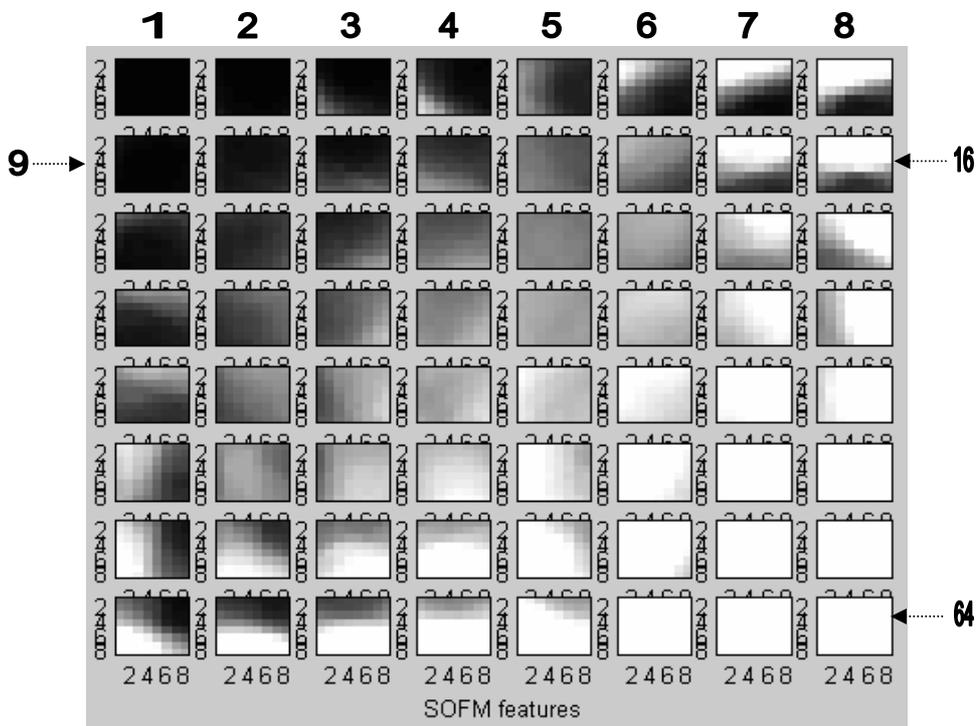
Fig(3) Local features extracted by MBCL



Fig(4) Local features extracted by MCL



Fig(5) Local features extracted by CL



SOFM Fig(6) Local features extracted by SOFM

5 Conclusions

From the results presented , one can see that the property of the learning in artificial neural networks effectively deals with problems of extracting appropriate information from an image relevant to the desired classification.

When biases were used in MBCL technique, all competitive layer neurons became active, the low-frequencies data representing the shape of a feature has been stored in the weights of each neuron, and high –frequencies details has been discarded. Inactive neurons and high-frequencies details could be seen in when CL and MCL techniques were used respectively.

Best recognition rate could be resulted when SOFM technique is used to extract features, while speedy system could be achieved when MBCL technique is used to do that. That is because, the lower the number of neurons, the smaller the time required for simulation.

6 References

- [1]C. Strouthopoulos, N. Papamarkos, C.Chamzas "PLA using RLSA and a neural network", Engineering Application of Artificial Intelligence, vol. 12, pp. 119-138,1999.
- [2] Sh. A. Dawwd "Face Recognition using Neocognitron Neural Network" M.Sc thesis, Univ of Mosul, Mosul, Iraq, July 2000.
- [3] I. Morns, S. Dlay, " The DSFPN, a New Neural Network for Optical Character Recognition", IEEE Transaction on Neural Networks, vol. 10, no.6, pp.1465-1473 (November 1999).
- [4]H. Ritter, G. Heidemann, "A neural 3-D object recognition architecture using optimized Gabor filters", Proceeding of 13th International conference on patter recognition , IEEE Computer Society Press,IV, pp.70-74, 1996.
- [5] S.V. Kartalopoulos, "Understanding Neural Networks and Fuzzy Logic" IEEE Neural Network Council, Sponsor Richared Saeks, NNC Laiison to IEEE Press, 1996.
- [6] K. J. Cois, I. Shin, "Image recognition neural network", Neurocomputing, vol. 7, pp. 159-185, 1995.
- [7] H. Demuth, M.Beale, "Neural Network Toolbox for use with Matlab", The MATHWORKS Inc, MA, USA, 1998.
- [8] K.Fukushima, "Neocognitron: A self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", Boil. Cybenet. Vol. 36, No. 4, pp. 193-202, April 1980.