Regular paper

Generated Moment Invariant Features by Cascaded Neural Network for Pattern Classification

PARAMESRAN RAVEENDRAN† and SIGERU OMATU††

This paper presents a technique on how to improve the intraclass invariance of moment invariant features of noisy images. Moment invariant features possess poor intraclass invariance in the presence of noise. Instead of using computational techniques of extracting moment invariant features from images, we use a trained feature extractor neural net to generate the second and third order moments. The generated moments are used as inputs to a trained classifier neural net which identifies the class the generated moments belong to. Noiseless and noisy binary images of numerals of 32× 32 matrix which have been translated, scaled, and rotated are used to determine the feasibility of the above technique. The zero-order regular moments are used in normalizing the binary images. The quality of generalization of the feature extractor neural net on the intraclass invariance of an image is examined. This is done by requiring each individually generated moments to fall within a tolerance bandwidth for the given pattern to be correctly classified. In addition to this, computed moments of normalized and unnormalized binary images are used as inputs to a single neural net to compare the effectiveness of this technique with respect to using generated moments. Back-propagation learning algorithm is used in the training of neural net. The number of hidden units used in the hidden layer of the classifier neural net is between 10 and 100. The cascaded neural net performs much better than using computed moments as inputs to a single neural net that functions as a classifier.

1. Introduction

The invariant properties of moments of 2D and 3D shapes have received considerable attention in recent years. Hu[1] published the first significant paper on the use of moment invariants for two-dimensional pattern recognition applications. He derived a set of invariants based on combinations of regular moments using algebraic invariants, which has the desirable properties of being invariant under image translation, scaling, and rotation. Moments and functions of moments have been utilized as pattern features in a number of applications.[1)~5)]

The features used in any pattern classification problems must posses large interclass separation and small intraclass invariance-slightly different shapes with similar general characteristics should have numerically close values. Moment invariant features posses good interclass separation and intraclass invariance under noise-free conditions. However, in the presence of noise they become unreliable.[6),7)] Conventional techniques and computational methods are applied to extract features in a preprocessing step, followed by neural net pattern classification.[8)]

In our technique using a cascaded neural net as shown in **Fig. 1**, we are proposing the first neural net to function as a feature extractor. This net will be trained to generate second and third order moment invariant features of noiseless binary images. The feature extractor neural net has to be trained with all possible combinations of the input pattern to achieve generalization. It is not possible to train a neural net to recognise shapes at every possible positions in a image plane, as the number of different patterns rapidly exceeds the capacity of the net to store exemplars. One approach to overcome this problem is to normalize the binary images before presenting to the network. The question is how many training examples are needed to be presented so that the network is able to properly

† Department of Electrical Engineering, University of Malaya
†† Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima

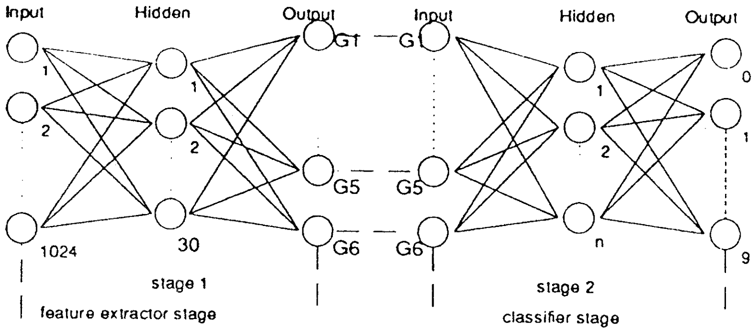


Fig. 1 Cascaded neural net.

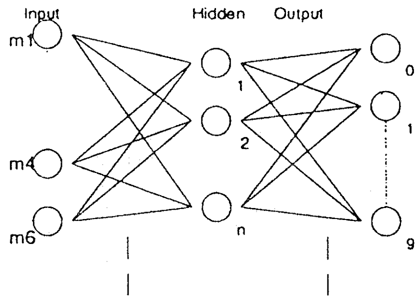


Fig. 2 Single neural net as a classifier.

generalize to new cases. An initial arbitrary value is used in training the feature extractor neural net.

After the network has been trained, the six outputs (generated moments) generated by the feature extractor neural net are tested by 240 unseen binary images which are representative of the data set for the quality of its generalization. A new technique is used in classifying the unseen pattern. We calculate the tolerance bandwidth for each generated moment and for a pattern to be classified correctly all the outputs must lie within the boundary of the lower and upper target values of the class the unseen pattern belongs to. If this number is less than a threshold value then the feature extractor is given more training examples.

Three sets of data sets are determined. The first set consists of computed moment invariant features of unnormalized binary images. The second set consists of moment invariant features of normalized binary images. Their features are used by a single classifier neural net, as shown in **Fig. 2**, which is trained to classify them. The third set consists of generated moment invariant features which are used as inputs to the classifier of the cascaded neural net. The performance of both the single and the cascaded neural nets are examined under different Signal to Noise Ratio (SNR).

The organization of the paper is as follows. Section 2 discusses the moment invariant features proposed by Hu.[1] Section 3 describes the neural net architecture and the back-propogation learning algorithm. Section 4 reports the experimental results on a 10-class data set consisting of numerals and also the generalization of the feature extractor neural net. Section 5 addresses the conclusion of the study.

2. Moment Invariant Features

Hu[1] has shown from these moments the second and third order invariant moments can be determined which are invariant not only to translation and scale variations, but also to rotation of image patterns.

$$m_{pq}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x^p y^q f(x, y)dxdy,$$

$$\text{for } p, q=0, 1, 2, \cdots \qquad (1)$$

Central moments which are invariant to translation can be defined as:

$$\mu_{pq}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^p(y-\bar{y})^q f(x, y)dxdy$$

where

$$\bar{x}=\frac{m_{10}}{m_{00}}, \qquad \bar{y}=\frac{m_{01}}{m_{00}}. \qquad (2)$$

Central moments can be normalized to scale invariant by

$$\eta_{pq}=\frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

where

$$\gamma=\frac{p+q}{2}+1. \qquad (3)$$

A set of nonlinear functions which are invariant to translation, scale and rotation can be expressed as:

$$m1 = \eta_{20} + \eta_{02}$$
$$m2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$m3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$m4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$m5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$m6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (4)$$

Regular and complex moments have been used in transforming an image to a predetermined standard form. Using moments through second order an image can be transformed into a standard form.[9] The number of training patterns needed to train the feature extractor neural net will be greatly reduced by transforming the image to a standard form.

In this paper, the zero-order regular moment, m_{00}, is used in transforming the image. Normalized binary images are used as inputs to the feature extractor neural net. The binary images are normalized with respect to scale and translation. Scale invariancy is achieved by enlarging or reducing each object such that the zero-order regular moment, m_{00}, is set to a predetermined value. Let $f(x/\alpha, y/\alpha)$ represent a scaled version of the image function, $f(x, y)$. Then the regular moment m_{pq} of $f(x, y)$ and m'_{pq} the regular moment of $f(x/\alpha, y/\alpha)$, are related by

$$m'_{pq} = \alpha^{p+q+2} m_{pq}. \quad (5)$$

The total number of object pixels in a standard image is determined from the arithmetic mean of the area of the training images and this is represented by m'_{00}. The relationship between the standard image and the unknown image, m_{00}, can be represented as $m'_{00} = \alpha^2 m_{00}$ where α is the scaling factor.

An image function, $f(x, y)$, can be normalized with respect to scale and translation by transforming it into $g(x, y)$, where

$$g(x, y) = f\left(\frac{x}{\alpha} + \bar{x}, \frac{y}{\alpha} + \bar{y}\right). \quad (6)$$

with $(\bar{x}, \bar{y})$ being centroid of $f(x, y)$ and can be computed using Eq. (2).

3. Neural Network

Artificial neural networks are the basis of a new computational technique inspired by the way the human brain works. Artificial neural networks with their inherent parallelism are ideally suited to image processing. **Figure 3** shows a three-layered feedforward network. The objective here is to find proper weights for all the connections such that a desired output is generated for corresponding input. The back-propagation learning algorithm is used in determining the ideal weights of all connections.

The back-propagation learning algorithm[10] involves a forward-propagation followed by a back-propagation step. Rather than developing a program to undertake a task, the network is trained by "learning from example." Both the forward and back-propagation steps are done for each pattern presentation during training.

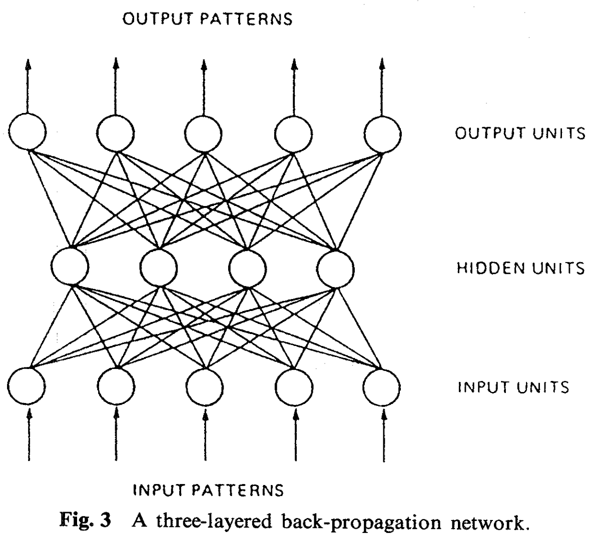


Fig. 3 A three-layered back-propagation network.

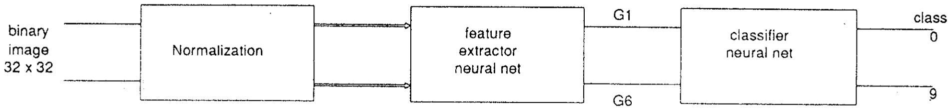


Fig. 4 Block diagram of normalized image using cascaded neural net.

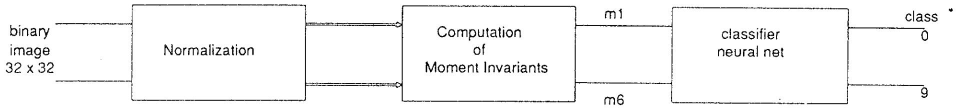


Fig. 5 Block diagram of normalized image using single neural net.

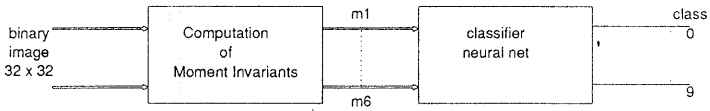


Fig. 6 Block diagram of unnormalized image using single neural net.

The network is presented with pairs of input pattern paired with target outputs. The input data is fed into neural net to produce its own output data and then this is compared with the desired output, resulting in an error signal for each output unit. Weights are adjusted to decrease the difference between the network's output and the target output. The patterns in the training set were presented to the neural net repeatedly until the Root Mean Square (RMS) value was 0.008.

Figure 4 shows the block diagram of the normalized image using two cascaded three-layered feedforward network that was used in our study. The first neural net (feature extractor neural net) of a cascaded two three-layered feedforward neural net as shown in Fig. 1, when on presentation of a binary image will generate the moment invariant features. These features will then be used by the second neural net, which acts as a classifier, will assign the class to which these features belong to. **Figure 5** shows the block diagram of computed moments of a normalized image classified by a neural net. **Figure 6** shows the block diagram of the computed moments of the unnormalized binary image classified by a neural net. Back-propagation learning algorithm is used in training the neural nets.

4. Experimental Study

In this section, the performance of a single and cascaded neural net to classify the 10 numerals are reported. In the case of the single neural net only computed moment invariant features of normalized and unnormalized binary images are used and the cascaded neural net uses moments that are generated by the feature extractor neural net.

Two sets of training data were used. The first set uses only 200 noiseless moment features as training examples and the performance of both the single and cascaded neural nets are examined under noiseless and noisy conditions. The second set consists of 400 noiseless moment features as training examples and is tested with the remaining data set. The performance of both the single and the cascaded neural nets are examined under different SNR.

4.1 The data sets

Four slightly different types of noiseless binary image of numerals 0 to 9 of 32×32 are written onto a board and the images are captured using a camera together with a frame grabber card. A thresholding value is used in converting the grey scale images to binary images. Each noiseless binary image of a numeral is then scaled, rotated, and translated to produce 56 noiseless binary images. **Figure 7** shows an example of a noiseless binary image of a numeral 9 that has been scaled, rotated, and translated. Each binary image of a numeral is perturbed with random noise. The SNRs of 50,

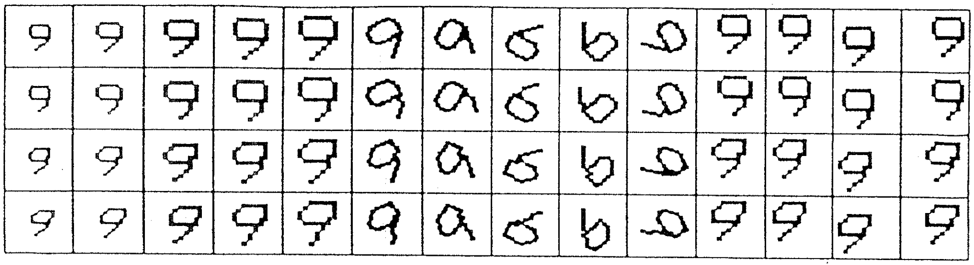


Fig. 7 The 56 scaled, rotated and translated images of numeral 9 in the data set.
Note the slight variations in shape shown in the first column.

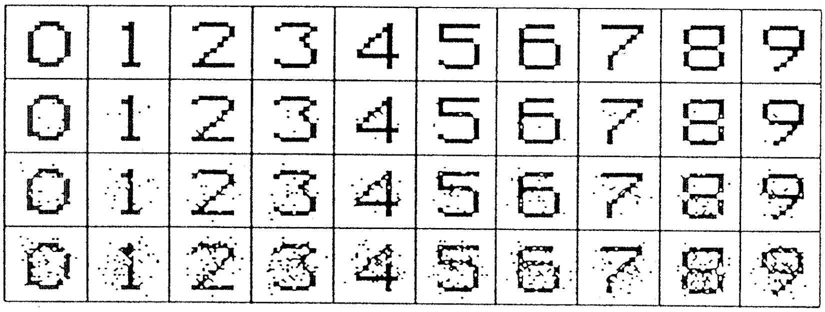


Fig. 8 Numerals from 0 to 9 with different levels of noise.
From top to bottom, SNR is noiseless, 50, 40, and 30 dB.

40, and 30 dB are used in this study. The total number of noiseless and noisy binary images of a numeral used is 224. **Figure 8** shows numerals 0 to 9 with different SNR's.

The noisy binary image is generated by randomly selecting some of the 1024 pixels of the noiseless binary image and reversing their values from 0 to 1 or vice versa. This random pixel selection is done according to a uniform probability distribution between 1 and 1024. The SNR is computed using $20 \log[(1024-N)/N]$, where N is the number of pixels which are different between a noisy image and a noiseless image.

Three sets of second and third order moment features of each binary image set of each numeral are determined. The first set was determined by computing each normalized binary image of each numeral using the equations in Eq. (5). The second set is determined just like the first set but in this case unnormalized binary images are used instead.

The third set is determined from the feature extractor neural net which is trained to generate second and third order moment features. After the neural net has been trained, the sets of noiseless binary images and noisy images of SNR 50, 40, and 30 dB are fed as inputs to the neural net. The neural net then generates the moment invariant features for each binary image that is presented to it.

In addition to that, 240 noisy binary images of all classes are simulated. These images are used to test the feature extractor neural net's ability to generalize after training.

4.2 Description of experiments

Two sets of training examples are used in this experiment to determine the performance of the respective classifiers. The training examples used in both the single and cascaded neural nets are of the same class of binary images. The data set of computed moments of unnormalized binary images, computed moments of normalized binary images, and generated moments is used in the training of the respective classifiers. Computed moments of normalized and unnor-

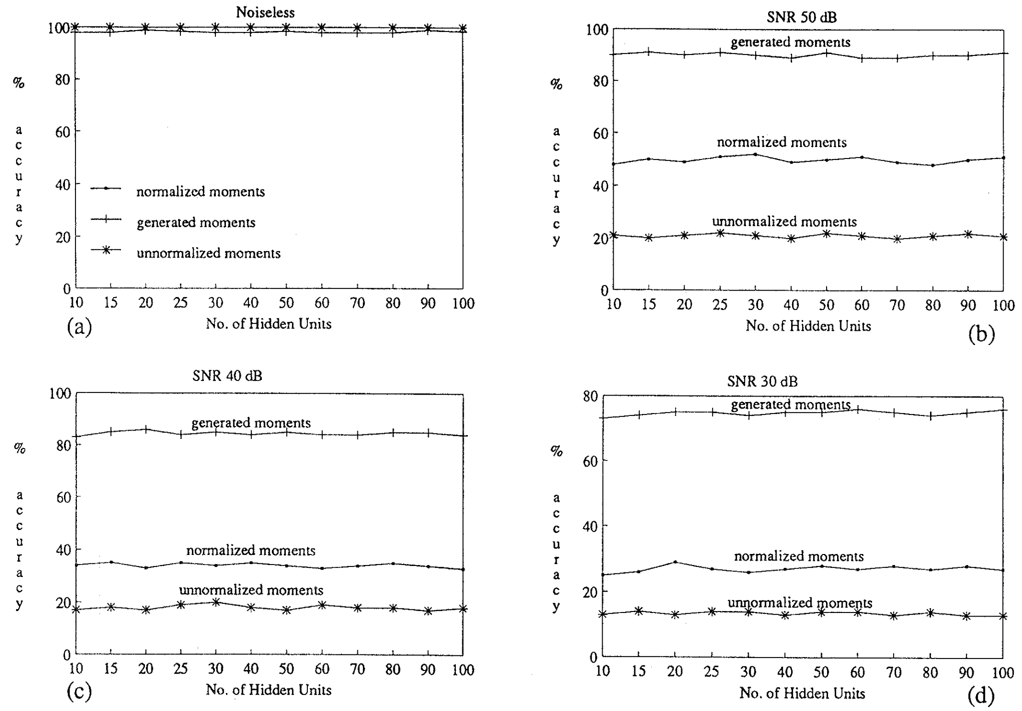


Fig. 9 Classification results using 200 images.

malized binary images are used as inputs to the single neural network. The cascaded neural network consists of two neural network, the first functions as a feature extractor neural network and the second neural network as a classifier. The feature extractor neural network generates the moment invariant features when presented with binary images, and these generated moments are used as inputs to the classifier, which assigns to which class they belong to.

In the first experiment we use only moment features of 200 noiseless binary images and is tested with noiseless and noisy images. **Figure 9** (a)-(d) shows the performance of the classifiers when trained with 200 samples of moment features of noiseless binary images.

In the second experiment, 400 training examples of moment features of the noiseless binary images are used and is tested with the remaining noiseless and noisy images. The training samples are presented in a random order. The training features are normalized to have zero mean and unit variance. This is necessary to ensure that a subgroup of the features does not dominate the weight adjustment process during training. The mth feature is normalized by

$$\bar{t}_m = \frac{t_m - \bar{t}_m}{\sigma_{tm}} \qquad (7)$$

where $\bar{t}_m$ and σ_{tm} are the sample mean and standard deviation of the mth training features of all classes. The neural net has converged to the error limit of 0.008 in less than 400 iterations when presented with generated moment invariant features. This is because the moment features of the same class typically vary little when perturbed with noise. After the network has learned and if one of the example from the training set is presented to the network it will generate 0.9 and above for the selected bit and the remaining bits are set to less than 0.1.

A continuing question in neural network research is the size of network needed to solve a particular problem. We used hidden layer nodes between 10 and 100 and find that 20 hidden layer nodes give the best or very close to the classification accuracy in all examined cases. **Figure 10** (a)-(d) shows the performance of the cascaded neural net and a single neural net when the number of hidden units are varied.

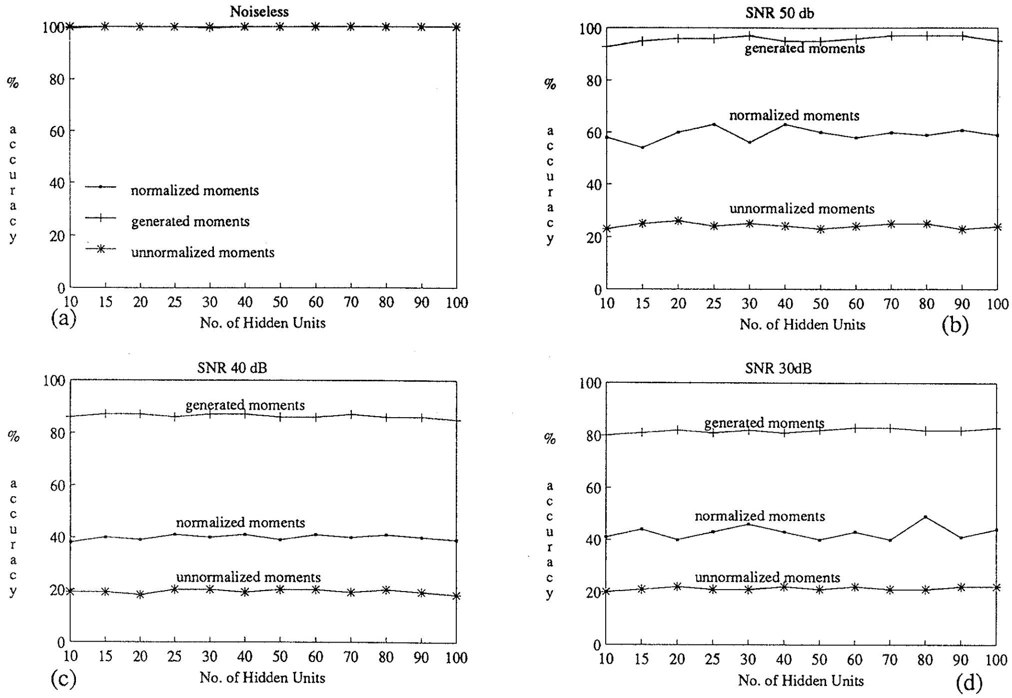


Fig. 10 Classification results using 400 images.

4.3 Generalization of feature extractor neural net

The feature extractor neural net is trained to generate second and third order moments. Noiseless and noisy images are used as input patterns and the target outputs are those of computed moments of noiseless binary images. We consider the network to have learned if the error between the target output and the network output is minimal. **Table 1** shows the standard deviation of the errors of the generated moments from the desired outputs of classes 0 to 9 for SNR 50 dB. **Table 2** shows the standard deviation of the errors of the computed moments from the computed moments of noiseless image of classes 0 to 9 for SNR 50 dB. The intraclass invariance of the generated moments is much better than computed moments of SNR 50 dB.

There is no clear indication as to the number of training patterns needed to train the network to achieve generalization, so we start off with 250 samples. After training, the network is tested for its ability to generalize. We present 240 unseen patterns which are representative of the data set to the network to generate the six moments.

For the effective classification of the unseen patterns, we have developed the tolerance level or bandwidth approximation for each individually generated moments. If δ_{jk} denotes the tolerance level, then

$$M_{jk}-\delta_{jk}< G_{jk}< M_{jk}+\delta_{jk} \qquad (8)$$

represents the interval in which G_{jk} should lie in order for the pattern to be correctly classified, where M_{jk} is the target output (moment) and G_{jk} is the generated output (moment), j represents the class, and k represents the component of the class. The tolerence interval can be computed and its formulation is given in Eq. (9). If the generated moments for each pattern falls within the range allowed for each class, then the pattern is classified correctly.

We next evaluate the deviations (G_k-M_{jk}) for each class j and obtain the maximum and minimum deviations. Let $M_{\max}$ and $M_{\min}$ denote the maximum and minimum deviations for any class j respectively. Then the difference between $M_{\max}$ and $M_{\min}$ gives the separation error within which lies all the other errors. The magnitude of the tolerance, δ_{jk}, is given by the following

Table 1 Standard deviation of the error of the generated moments of all classes for SNR 50 dB.

	Generated moments					
Class	G_1	G_2	G_3	G_4	G_5	G_6
0	0.0135	0.2733	0.1425	0.1779	0.2361	0.3129
1	0.0093	0.0272	0.1480	0.0907	0.1186	0.0932
2	0.0161	0.0672	0.3259	0.1224	0.2352	0.1459
3	0.0294	0.0729	0.2540	0.0863	0.1217	0.1171
4	0.0286	0.0682	0.4930	0.1665	0.2781	0.1754
5	0.0132	0.0353	0.1062	0.0437	0.0750	0.0920
6	0.0195	0.0728	0.4483	0.1399	0.2985	0.1724
7	0.0120	0.0308	0.0876	0.0568	0.1251	0.1357
8	0.0290	0.1054	0.1877	0.1070	0.1062	0.1571
9	0.0247	0.0442	0.3052	0.0831	0.0944	0.0768

Table 2 Standard deviation of the errors of the computed moments of all classes for SNR 50 dB.

	Computed moments (Normalized images)					
Class	M_1	M_2	M_3	M_4	M_5	M_6
0	0.0465	0.5287	1.2744	0.7400	1.4465	1.0511
1	0.0184	0.0657	1.5696	0.9629	1.7560	0.6972
2	0.0330	0.0875	0.9226	0.4160	0.7759	0.4014
3	0.0387	0.1808	0.8561	0.3439	0.8530	0.4954
4	0.0397	0.1429	0.7289	0.4139	0.8956	0.8543
5	0.0385	0.1087	1.0693	0.4645	0.8077	0.4444
6	0.0340	0.1883	0.4607	0.3542	0.6696	0.4484
7	0.0354	0.0071	0.2551	0.1340	0.4632	0.2962
8	0.0356	0.1577	1.0575	0.5766	1.1332	0.6054
9	0.0347	0.0969	0.5779	0.2037	0.4307	0.3894

equation.

$$\delta_{jk} \propto \frac{|G_k - M_{jk}|}{\sum_{j=0}^{9} |G_k - M_{jk}|} \cdot |(M_{\max} - M_{\min})|. \quad (9)$$

When the inequality as given by the Eq. (8) holds true for all admissible values of $k(k=1$ to $6)$ then the pattern will correctly belong to class j. We test the generalization ability of the feature extractor neural net by measuring the number of patterns it is able to classify accurately.

5. Conclusion

A neural net containing a feature extractor trained to generate second and third order moments and a classifier has been used for the recognition of noisy and noiseless binary images. Moment invariant features possess good large interclass separation but rather poor intraclass invariance-images of the same class with slight change in image content. Using a feature extractor neural net it is possible to train the network to minimize the intraclass invariance.

The distinct advantage of using a feature extractor neural net is that it attempts to generate outputs that it has learned during training when presented with images that it has not seen. We found the network was able to generate outputs close to target outputs for an image (still recognizable) whose boundary was distorted. As the distance of its noise points from the centroid of an image increases, the computed moments varies substantially, and to classify them to the correct class is a difficult task. In the case of the feature extractor neural net it does not behave in the same manner, since it is trained to generate outputs close to the target outputs. This helps to improve the performance of the classifier.

In this paper, to test the generalization of the feature extractor neural net, we developed the tolerance level or the bandwidth approximation for each individually generated moments. If the proximity of the generated outputs with respect to the target outputs are to be maintained for any given input variations within an acceptable range, then the differences amongst the weights between the input and hidden layer of the feature extractor neural network should be small.

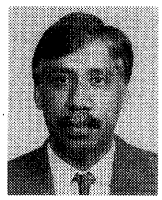
The performance of the cascaded neural net to classify noisy images of SNR 50 dB is about 90% and SNR 40 dB is about 80% after being trained with generated moments of noiseless binary images. The performance improves when trained with 400 images of generated moments. The performance by computed moments of normalized binary images was far better than that of computed moments of unnormalized binary images, but not as good as using generated moments.

Further work should be explored on the significance[11] of each of the features used in classifying an image and also which combination of feature is sufficient to classify the patterns correctly.

References

1) Hu, M. K.: Visual Pattern Recognition by Moment Invariants, *IRE Trans. Information Theory*, Vol. IT-8, No. 2, pp. 179-187 (1962).
2) Teague, M. R.: Image Analysis via General Theory of Moments, *Journal of Optical Society of America*, Vol. 70, No. 8, pp. 920-930 (1980).
3) Smith, F. W. and Wright, M. H.: Automatic Ship Photo Interpretation by the Method of Moments, *IEEE Trans. Comput.*, Vol. C-20, No. 9, pp. 1089-1094 (1971).
4) Boyce, J. F. and Hossack, W. J.: Moment Invariants for Pattern Recognition, *Pattern Recognition Letters*, No. 5-6, pp. 451-456 (1983).
5) Reddi, S.: Radial and Angular Moment Invariants for Image Identification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 2, pp. 240-242 (1986).
6) Teh, C. H. and Chin, R. T.: On Image Analysis by the Methods of Moments, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-10, No. 4, pp. 496-513 (1988).
7) Raveendran, P., Omatu, S. and Wong, W. Y.: Performance of Generated Moments by Neural Network to Recognise Invariant Image Representation, *IJCNN*, Vol. 1, No. 3, pp. 689-694, Beijing, China (1992).
8) Khotanzad, A. and Lu, J. H.: Classification of Invariant Image Representations Using a Neural Network, *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-38, No. 6 (1990).
9) Abu-Mostafa, Yaser S. and Psaltis, Demetri: Image Normalization by Complex Moments, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 1, pp. 46-55 (1985).
10) Rumelhart, D. E. and McClelland, J. L.: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge, Massachusetts (1986).
11) Fukunaga, K.: *Introduction to Statiscal Pattern Recognition*, Academic, New York (1972).

(Recieved April 30, 1993)
(Accepted November 11, 1993)



Paramesran Raveendran was born in Ipoh, Malaysia on May 8, 1958. He received B.Sc. and M.Sc. degrees in electrical engineering from South Dakota State University, Brookings, South Dakota in 1984 and 1985, respectively. Upon graduation he worked as a system designer at Daktronics, South Dakota and then joined University of Malaya in 1986 where he is a Lecturer at Department of Electrical Engineering. He has been pursuing his Ph.D. at Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima, Japan under JSPS Ronpaku (Dissertation Ph.D.) program since 1992. His current interest centers on pattern recognition by neural networks.

Sigeru Omatu was born on December 16, 1946. He received B. E. degree in electrical engineering from University of Ehime, Japan. in 1969, and M.E. and Ph.D. degrees in electronics engineering from University of Osaka Prefecture, Osaka, Japan, in 1971 and 1974, respectively. From 1974 to 1975 he was a Research Associate and in 1988, a Professor at the University of Tokushima, Japan. From November 1980 to February 1981 and from June to September 1986, he was a Vising Associate of Chemical Engineering Department at California Institute of Technology, Pasadena. From November 1984 to February 1985 he was a Visiting Researcher at International Institute for Applied Systems Analysis, Austria. His current interests center on neural networks and control system theory. Dr. Omatu received the best paper award from IEE of Japan in 1991. He is Associate Editors of the IMA Journal of Mathematical Control and Information (U.K.) and IEEE Transactions of Neural Networks (USA). He is coauthors of Distributed Parameter Systems: Theory and Applications (Oxford University Press) and Neuro-Control and its Applications (Springer-Verlag). He is a member of IPSJ.