NEURAL NETWORK-BASED SEGMENTATION OF TEXTURES USING GABOR FEATURES

A. G. Ramakrishnan, S. Kumar Raja, and H. V. Raghu Ram
Dept. of Electrical Engg., Indian Institute of Science
Bangalore - 560 012, India
E-mail: ramkiag@bhaskara.ee.iisc.ernet.in

Abstract. The effectiveness of Gabor filters for texture segmentation is well known. In this paper, we propose a texture identification scheme, based on a neural network (NN) using Gabor features. The features are derived from both the Gabor cosine and sine filters. Through experiments, we demonstrate the effectiveness of a NN based classifier using Gabor features for identifying textures in a controlled environment. The neural network used for texture identification is based on the multilayer perceptron (MLP) architecture. The classification results obtained show an improvement over those obtained by K-means clustering and maximum likelihood approaches.

INTRODUCTION

The real world is abound with multiple textures combined in an arbitrary fashion. To this day, the problem of identifying various textures in a natural image has not been completely solved. The two most important issues in any texture segmentation or classification algorithm which play an important role in the accuracy of classification are the features and the decision function used for classification. In this regard, neural networks have demonstrated to be robust and accurate decision functions. Neural networks have been very successfully applied in various fields including satellite image data classification. In this paper, we have explored the possibility of using the multilayer perceptron (MLP) for classifying textures. Kolmogorov and others have proved that any decision function can be implemented to any degree of arbitrary accuracy using a multilayer perceptron.

Our goal is to identify textures present in a controlled environment where the kind of textures present is already known. For instance, consider an environment in which only textures of the kind shown in Fig. 3 occur. In 10, we have shown a mosaic consisting of textures shown in Fig. 3. Our objective is to identify the textures present in the mosaic (Fig. 10). To this end, we propose a supervised scheme for classifying textures using a MLP.

GABOR FEATURES FOR TEXTURE CLASSIFICATION

For any classification scheme (supervised or unsupervised), the choice of features is very crucial. In this work, we have chosen the Gabor features for texture classification. A number of researchers have already demonstrated the effectiveness of the Gabor features for discrimination of textures. The main advantage of the Gabor filters is that they provide optimal simultaneous resolution in both space and frequency domains. The expressions for discrete 2D Gabor cosine and sine filters are given by

$$g_c[i,j] \quad = \quad B\exp(-\frac{i^2+j^2}{2\sigma^2})\cos(2\pi f(i\cos\theta + j\sin\theta)) \qquad (1)$$

$$g_s[i,j] \quad = \quad C\exp(-\frac{i^2+j^2}{2\sigma^2})\sin(2\pi f(i\cos\theta + j\sin\theta)) \qquad (2)$$

where B and C are normalizing factors to be determined. In this paper, we consider filters having equal spread in both i and j directions. The response of the filter is determined by the spread parameter σ of the Gaussian and the radial frequency f of the modulating sinusoid. The orientation of the Gabor filter is determined by the parameter θ. B and C are chosen such that they satisfy the following conditions

$$\sum_i\sum_j g_c[i,j]\cos(2\pi f(i\cos\theta + j\sin\theta)) = 1 \qquad (3)$$

$$\sum_i\sum_j g_s[i,j]\sin(2\pi f(i\cos\theta + j\sin\theta)) = 1 \qquad (4)$$

Since the filters have the same spread in both i and j directions, the contours on which the magnitude of the Fourier transform is constant are circles. The important issue in the design of Gabor filters for texture classification is the choice of filter parameters; radial frequency f, the spread of the filters σ and the orientation of the filter θ. For the purpose of texture classification, we design twenty five Gabor filters; six orientations, four filters per orientation and one low frequency filter. The six orientations of the Gabor filters are $0^o, 30^o, 60^o, 90^o, 120^o$ and 150^o. The radial frequencies and the spread of the four Gabor filters for each orientation are given in Table 1. The radial frequency of the low frequency filter is zero and its spread is 3.464907.

Angle of orientations	radial frequency	spread of the filter
0^o, 30^o, 60^o 90^o, 120^o, 150^o	0.072968	9.922465
	0.123928	5.842255
	0.210480	3.429866
	0.357478	2.025361

Table 1: Radial frequency and spread of the Gabor filters

In Fig. 1, we have shown the response of twenty five Gabor cosine filters designed using the parameters given in Table 1. The circles in Fig. 1 represent regions where the magnitude of the Gabor filters is greater than half the peak magnitude. The same parameters of the Gabor cosine filters are used to design twenty five Gabor sine filters. The procedure used to compute the filter parameters is beyond the scope of this paper.

Let the size of the Gabor cosine and sine filters be $(2K+1) \times (2L+1)$. The inner products of the Gabor cosine and sine filters with the input image x at the location (i,j) are given by

$$R_c[i,j] = \sum_{k=-K}^{K} \sum_{l=-L}^{L} x[i+k,j+l] \cdot g_c[k,l] \tag{5}$$

$$R_s[i,j] = \sum_{k=-K}^{K} \sum_{l=-L}^{L} x[i+k,j+l] \cdot g_s[k,l] \tag{6}$$

At each pixel location (i,j), the magnitude of the vector with $R_c[i,j]$ and $R_s[i,j]$ as components is computed as

$$R[i,j] = \sqrt{R_c^2[i,j] + R_s^2[i,j]} \tag{7}$$

So at each location, we have twenty five outputs according to Eqn. 7 corresponding to the twenty five filters we have designed. These outputs are used as a twenty five dimensional feature vector for classification by the MLP.

MULTILAYER PERCEPTRON

A MLP consists of layers of structurally identical computing nodes (neurons) arranged so that the output of every neuron in one layer feeds into the input of every neuron in the next layer. See Fig. 2 for a typical topology of a MLP. The first layer is referred to as the input layer and the last layer is referred to as the output layer. The layers between the input and the output layers are referred to as the hidden layers. The number of units (nodes) in the input layer is equal to the dimension of the feature vector. The number of units in the output layer equals the number of classes the neural network is required to recognize. The network recognizes a feature vector as belonging to class k if the output of the kth node is greater than that of the other nodes in the output layer.

The output of the nodes in the input layer is just the values of the components of the feature vector of the input pattern. For the other layers, the output of a node in one layer is related to the output of the nodes in the previous layer by the equations

$$y_j^{(M)} = f(v_j^{(M)} + b_j^{(M)}) \tag{8}$$

$$v_j^{(M)} = \sum_{i=1}^{N^{(M-1)}} W_{ji}^{(M)} y_i^{(M-1)} \tag{9}$$

where $y_j^{(M)}$ and $b_j^{(M)}$ are the output and the bias of the jth node in the Mth layer, respectively. f is an activation function, and $y_i^{(M-1)}$ is the output of the ith node in the $(M-1)$th layer. $N^{(M-1)}$ is the number of nodes in the $(M-1)$th layer, and $W_{ji}^{(M)}$ is the weight between the jth node in the Mth layer and the ith node in $(M-1)$th layer. We have chosen the activation function f to be the sigmoid function and it is the same for all the nodes.

$$f(v) = \frac{1}{1+\exp(-v)} \tag{10}$$

Training of the MLP

A supervised scheme for texture classification using a MLP requires the MLP to be trained with a few examples of textures. In this regard, we consider only a set of nine textures shown in Fig. 3. The textures are given unique indices for identification. For a mosaic of textures, a texture class map is defined as a matrix which contains the index of texture in each pixel location of the mosaic. For instance, for the mosaic shown in the Fig. 3, the texture class map is shown in Fig. 4.

Training of the MLP plays an important role in the performance of the neural network in classifying or identifying textures. In this paper, we have chosen a MLP with three layers: input, hidden, and output (see Fig. 1), for texture identification. Since the MLP needs to discriminate nine textures, the number of output units in the output layer is nine. Also, since we have chosen twenty five Gabor features for texture discrimination according to Eqn. 7, the number of units in the input layer is twenty five. There is no clear criteria for the choice of the number of units in the hidden layer. We have chosen the number of units in the hidden layer to be fifty.

For training the MLP, we compute the Gabor features according to Eqn. 7 for the image shown in Fig. 3 and randomly select 4100 Gabor feature samples from each of the textures. These 36900 samples constitute the training set. For each feature vector present in the training set, the desired output in the output layer is the texture index obtained from the texture class map shown in Fig. 4. For instance, if the texture index of the training feature vector is 6, then the desired output in the output layer is 000001000. The difference between the desired output and the computed output is the error measure. This value is used to update the parameters (weights and biases) of the network. The sequence of feeding the network with training feature vectors and using the error measure to update the parameters is continued till the error measure is minimized. The weight updation using the error measure is carried out by using the backpropagation algorithm. For details, see [9]. The main points to be noted are: (i) the order of presentation of training samples should be randomized from epoch to epoch; and (ii) the momentum and learning rate parameters are typically adjusted (and usually decreased) as the number of training iterations increases.

Texture Classification by the MLP

Once the training is completed, the neural network is used to classify the rest of the mosaic image shown in Fig. 3. For each pixel in the test image, the Gabor feature according to Eqn. 7 is computed and fed to the trained MLP. The output at the output layer of the network is computed using Eqn. 8. The texture class of the pixel is determined by the number of the node in the output layer which has the maximum output. If the maximum is less than a threshold or two nodes have the same maximum, then the class of the pixel is considered to be unknown. The accuracy of the classification of the test mosaic image by the neural network is found by comparing the output of the neural network with that of the corresponding texture class map.

IMPLEMENTATION AND RESULTS

All implementation has been carried out on mosaics containing subsets of the textures shown in Fig. 3. First we demonstrate the effectiveness of a MLP in discriminating two textures. We choose a mosaic of two natural textures shown in Fig. 5. In order to discriminate the two textures, we choose a MLP having two output units, fifty units in the hidden layer and twenty five units in the input layer. The twenty five units in the input layer correspond to the number of Gabor features we have chosen for texture discrimination. The MLP is trained according to the procedure described in earlier. The classification obtained by the trained MLP for Fig. 5 is shown in Fig. 7. The accuracy of the classification of the mosaic image in Fig. 5 by the neural network is found by comparing the output of the neural network with that of the texture class map which is shown in Fig. 6. We have compared the result of neural network-based segmentation with that obtained by K-means clustering. The result of clustering the Gabor features of the mosaic image in Fig. 5 is shown in Fig. 8. Clearly, the result of segmentation by the MLP is superior to that obtained by the clustering procedure.

In Fig. 10, we have shown a mosaic of two textures. The result of segmentation by the MLP is shown in Fig. 12. Similarly, in Fig. 15, we have shown a mosaic of four textures. The same MLP is used to identify the textures present in the mosaic and the result of segmentation is shown in Fig. 17. The result of clustering the Gabor features of the images shown in Figs. 10 and 15 are shown in Figs. 13 and 18, respectively. The black "spots" or "blobs" seen in the neural output corresponds to the pixels which are wrongly classified. Figs. 20 and 21 compare the results of segmentation of all the nine textures in Fig. 3 by the MLP and clustering.

In addition to K-means clustering, we have also compared the segmentation results of the neural network with that obtained from a statistical classifier such as Gaussian maximum likelihood (MAXL). For Fig. 5, the classification result obtained from the MAXL is shown in Fig. 9. Similarly for the Figs. 10 and 15, the MAXL segmentation outputs are shown in Figs.

14 and 19, respectively.

Table 2: Comparison of performance of MLP and MAXL

Figure No:	Classification accuracy of MLP	Classification accuracy of MAXL
3	97.94 %	90.01 %
5	98.66 %	97.73 %
10	95.91 %	95.05 %
15	91.92 %	87.35 %

In Table 2, we have given the accuracy of classification by the MLP and MAXL for the various mosaics. From the table, it is evident that a neural network-based classifier has distinct advantages over the conventional statistical classifiers such as Gaussian maximum likelihood. However, the use of neural networks (NN) for data classification may yield erroneous results due to over-fitting of data. In other words, in case the NN has very large number of hidden nodes and if the training of the neural network is continued for a very long period of time, then the NN fails to generalize from the given training examples successfully. The smallest neural network which yields the maximum classification accuracy during testing is considered to be the network which has the maximum generalization capability.

In order to overcome the problem of over-fitting, the best solution would be as follows.

1. Start with a neural network with a small number of hidden units.
2. Train and test the NN and compute its classification accuracy for various samples.
3. Increase the number of hidden units and perform the training and testing procedure.
4. As the number of hidden units increase, the classification accuracy of the network during testing increases and then may saturate or decrease. The smallest network which yields the highest classification accuracy is taken to be the one with the best generalization abilities.

Further, one can reduce the complexity of the neural network by resorting to techniques which remove redundancy in the input feature vector. One of the most popular techniques of feature dimensionality reduction is principal component analysis (PCA). Using PCA, along with the growing technique described above, it may be possible to obtain a robust texture classifier with the least complexity.

CONCLUSIONS

In this paper, we have proposed a neural network-based scheme for texture identification in a controlled environment using Gabor features. The neural network is trained using the backpropagation algorithm, and tested on different mosaics. The accuracy of classification is computed by comparing the output of the neural network with that of the texture class map of the mosaic. It has been found that MLP offers an accuracy of higher than 90 % in our experiments. The greater effectiveness of the MLP has also been established by comparing MLP classification results with those obtained from the K-means clustering and Gaussian maximum likelihood.

ACKNOWLEDGMENT

The authors acknowledge the valuable comments of the anonymous reviewers.

REFERENCES

[1] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel Texture Analysis using Localized Spatial Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 55-73, 1990.

[2] A. C. Bovik, "Analysis of Multichannel Narrow-band Filters for Image Texture Segmentation," *IEEE Transactions on Signal Processing*, Vol. 39, pp. 2025-2043, 1991.

[3] A. K. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation using Gabor Filters," *Pattern Recognition*, Vol. 23, pp. 1167-1186, 1991.

[4] A. C. Bovik, N. Gopal, T. Emmoth, and A. Restrepo, "Localized measurements of emergent image frequencies by Gabor Wavelets," *IEEE Transactions on Information Theory*, Vol. 38, pp. 691-711, 1992.

[5] D. F. Dunn and W. E. Higgins, "Optimal Gabor-filter Design for Texture Segmentation," *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. V, V37-V40, 1993.

[6] D. F. Dunn, W. Higgins, and J. Wakeley, "Texture Segmentation using 2-D Gabor Elementary Functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 130-149, 1994.

[7] D. F. Dunn and W. E. Higgins, "Optimal Gabor Filters for Texture Segmentation," *IEEE Transactions on Image Processing*, Vol. 4, pp. 947-964, 1995.

[8] J. Bigun and J. M. H. du Buf, "N-folded Symmetries by Complex Moments in Gabor Space and their Applications to Unsupervised Texture Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 80-87, 1994.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall International, 1999.

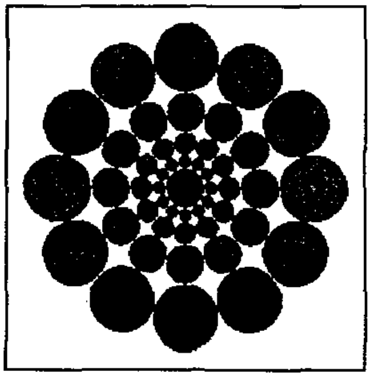

Figure 1: Response of the Gabor cosine filters

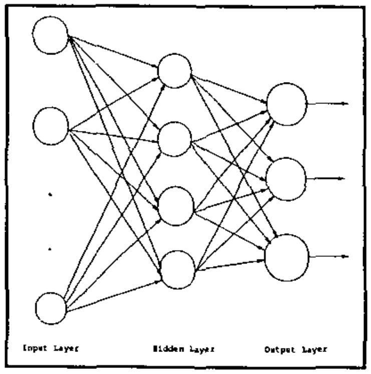

Figure 2: A typical topology of a multilayer perceptron network.

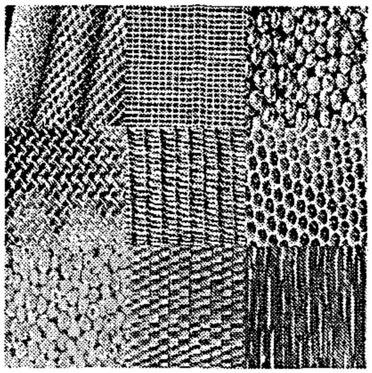

Figure 3: Set of nine textures

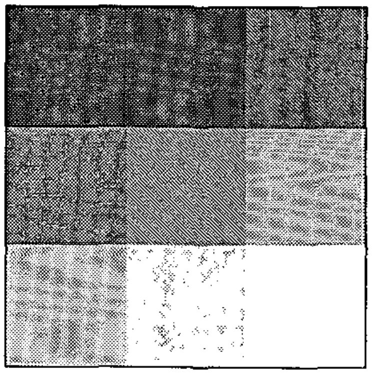

Figure 4: Texture class map of the mosaic shown in Fig. 3

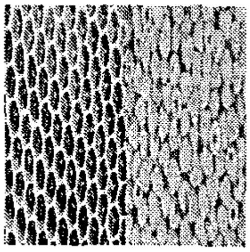

Figure 5: Mosaic of two natural textures

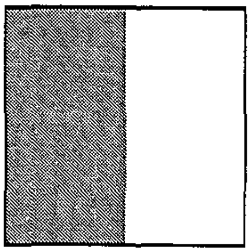

Figure 6: Texture class map of Fig. 5

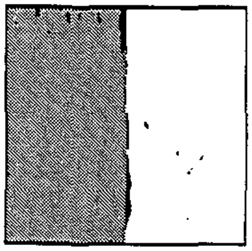

Figure 7: Classification result obtained using MLP

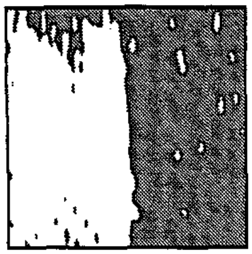

Figure 8: Classification result obtained using clustering

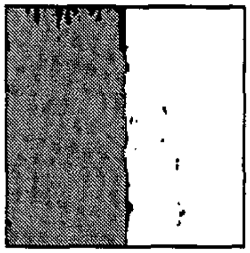

Figure 9: Classification result obtained using MAXL

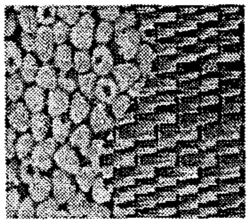

Figure 10: Mosaic of two natural textures present in Fig. 3

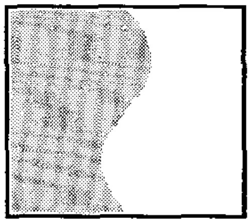

Figure 11: Texture class map of Fig. 10

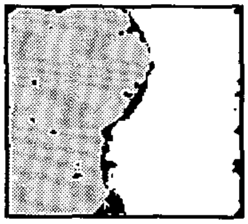

Figure 12: Classification result obtained using MLP

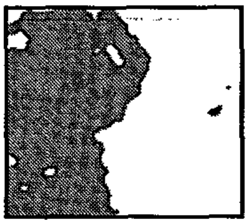

Figure 13: Classification result obtained using clustering

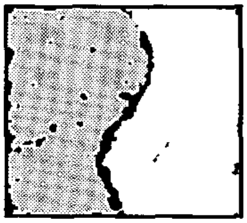

Figure 14: Classification result obtained using MAXL

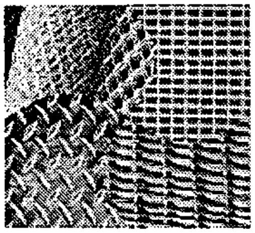

Figure 15: Mosaic of two natural textures present in Fig. 3

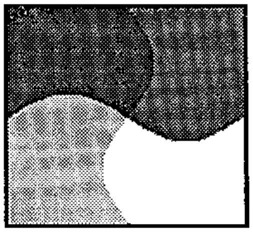

Figure 16: Texture class map of Fig. 15

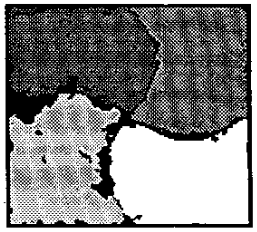

Figure 17: Classification result obtained using MLP

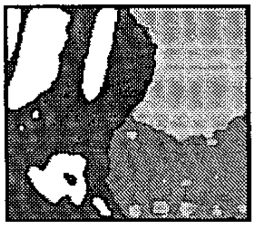

Figure 18: Classification result obtained using clustering

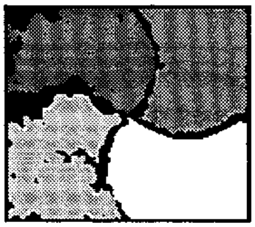

Figure 19: Classification result obtained using MAXL

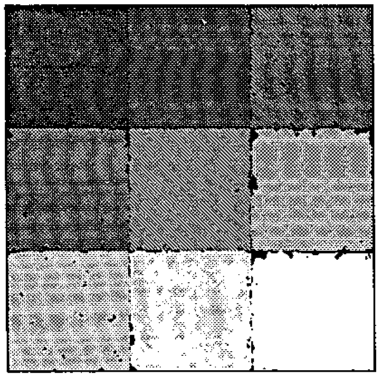

Figure 20: Classification result obtained by the neural network

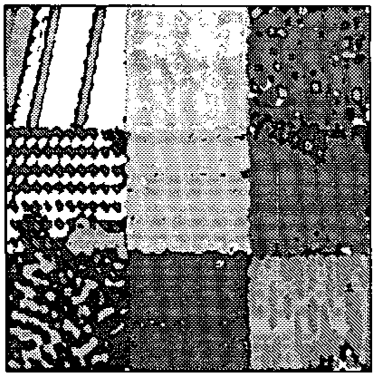

Figure 21: Classification result obtained by clustering