Texture Classification using Convolutional Neural Networks

Conference Paper · December 2006

DOI: 10.1109/TENCON.2006.343944 · Source: IEEE Xplore

2 authors:

Fok Hing Chi Tivive
University of Wollongong
55 PUBLICATIONS **593** CITATIONS
SEE PROFILE

Abdesselam Bouzerdoum
University of Wollongong
245 PUBLICATIONS **3,749** CITATIONS
SEE PROFILE

2006

Texture Classification using Convolutional Neural Networks

F. H. Tivive
University of Wollongong, fok-hing-chi_tivive@uow.edu.au

A. Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

Publication Details

This paper was originally published as: Tivive, FHC & Bouzerdoum, A, Texture Classification using Convolutional Neural Networks, 2006 IEEE Region 10 Conference (TENCON 2006), Hong Kong, China, 14-17 November 2006, 1-4. Copyright IEEE 2006.

Texture Classification using Convolutional Neural Networks

Abstract

In this paper, we propose a convolutional neural network (CoNN) for texture classification. This network has the ability to perform feature extraction and classification within the same architecture, whilst preserving the two-dimensional spatial structure of the input image. Feature extraction is performed using shunting inhibitory neurons, whereas the final classification decision is performed using sigmoid neurons. Tested on images from the Brodatz texture database, the proposed network achieves similar or better classification performance as some of the most popular texture classification approaches, namely Gabor filters, wavelets, quadratic mirror filters (QMF) and co-occurrence matrix methods. Furthermore, The CoNN classifier outperforms these techniques when its output is postprocessed with median filtering.

Publication Details

This paper was originally published as: Tivive, FHC & Bouzerdoum, A, Texture Classification using Convolutional Neural Networks, 2006 IEEE Region 10 Conference (TENCON 2006), Hong Kong, China, 14-17 November 2006, 1-4. Copyright IEEE 2006.

Texture Classification using Convolutional Neural Networks

Fok Hing Chi Tivive* and Abdesselam Bouzerdoum, *Senior Member, IEEE* †
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, Northfields Avenue
Wollongong, NSW 2522, AUSTRALIA.
*tivive@uow.edu.au, †a.bouzerdoum@ieee.org

Abstract— **In this paper, we propose a convolutional neural network (CoNN) for texture classification. This network has the ability to perform feature extraction and classification within the same architecture, whilst preserving the two-dimensional spatial structure of the input image. Feature extraction is performed using shunting inhibitory neurons, whereas the final classification decision is performed using sigmoid neurons. Tested on images from the Brodatz texture database, the proposed network achieves similar or better classification performance as some of the most popular texture classification approaches, namely Gabor filters, wavelets, quadratic mirror filters (QMF) and co-occurrence matrix methods. Furthermore, The CoNN classifier outperforms these techniques when its output is postprocessed with median filtering.**

I. Introduction

Texture analysis which consists of texture classification, segmentation, and synthesis has a wide variety of applications in image processing, e.g., industrial and biomedical surface inspection, ground classification and segmentation of satellite imagery, and content-based image retrieval. Texture, even though humans can effortlessly recognize texture, is very difficult to define. Many different texture definitions were reported in the literature, but still there is no common definition [1]. Despite numerous studies, texture classification has remained an elusive pattern recognition task. The fundamental issues in texture classification are how to extract compact features to represent the texture and what type of metric to use in comparing the feature vectors. The classical framework of texture classification is to transform the texture image into a feature vector using a bank of filters, followed by some nonlinearity and smoothing steps before classification. In [2], Randen and Husøy performed a comprehensive study on texture classification where several filtering approaches were evaluated.

Two early texture analysis techniques are the co-occurrence matrix [3] and Markov random fields (MRFs) [4], which have been popularly used for texture classification. However, the analysis of spatial interactions of these methods is constrained to a relative small neighborhoods, which may result in a limited expressive power [5]. More recently, advanced filter bank methods, such as Gabor and wavelet filters, have been developed for texture classification and segmentation [6]–[8]. One of the drawbacks of these approaches is that the generated filters usually discriminate a wide variety of textures, and this requires either an optimization method or manually select the filters for a given set of textures. Moreover, some of the optimization techniques are restricted to a two-class problem, or may be computationally intensive [2].

Jain and Karu [9] combined the texture feature extraction and classification in a unified framework by embedding the principles of multichannel filtering scheme into a neural network architecture. The first network they developed is a three layer (including the input layer) feedforward network, i.e., a *multilayer perceptron* (MLP), with each input node fully connected to a small region of size $M \times M$ in the input image. The second network is similar to that proposed by LeCun *et al.* [10], where the hidden neurons use a weight sharing mechanism to connect to the previous layer. The latter network has around 5000 weights and was trained with an online backpropagation algorithm. To reduce the number of weights and input filters, they applied a pruning algorithm developed by Mao *et al.* [11].

In this paper, we develop a convolutional neural network (CoNN) derived from our previous work [12] for texture classification. The network is based on the same structural ideas as LeNet-5 [13], but with a much simpler network structure and a systematic connection scheme, which results in a small number of free network parameters. For comparison purposes, the CoNN is evaluated on texture images taken from [2]. The next section describes the proposed CoNN structure and highlights its novel aspects, in comparison to its predecessor CoNN models. Section III presents the training process and the texture images used for testing. The experimental results and performance analysis is given in Section IV, and Section V presents some concluding remarks on the appearance-based method for texture classification.

II. Convolutional Neural Network Architecture

The CoNN that we have developed here for texture classification is a four layer network: one input layer, two hidden layer, and one output layer. The network receives inputs from a two-dimensional (2-D) array of size 13×13 pixels, and produces an output that indicates the texture class to which of the center pixel belongs. The arrangement and connection of the neurons in the hidden

1-4244-0549-1/06/$20.00 ©2006 IEEE.

layers are different from those of an MLP. Neurons in each hidden layer are arranged into planes, called *feature maps*, and each neuron is locally connected to a small neighborhood of the input plane. This connection strategy is derived from our understanding of the mammalian visual system, where ganglion cells are connected to groups of photoreceptors; the region from which a cell receives its inputs is known as the *receptive field*. In CoNN, the weights linking the neuron to its receptive field, including the bias term, are shared among all neurons in the feature map, which we refer to as the weight sharing mechanism. The receptive field sizes used in the first and second hidden layers are 7×7 and 5×5, respectively. In the first hidden layer, the feature maps are down-sampled by a factor of two; this is done by simply shifting the receptive field centers of adjacent neurons by two positions in the vertical and horizontal directions. Consequently, the network inherits some degree of invariance to geometric distortions since higher-order features are no longer dependent on their absolute positions in the input image. For the second hidden layer, no down-sampling operation is applied; hence the size of the feature maps is the same as those in the first hidden layer. To incorporate some smoothing effect on the extracted features, a local averaging operation is performed on all feature maps in the second hidden layer; that is, a non-overlapping mask of size 2×2 is used to average every four outputs into a single signal, which is fed to the output neurons. Each output neuron represents one texture class, and a winner-take-all technique is applied to classify the texture of the central pixel in the 13×13 input region.

In contrast to early models of CoNN, the proposed network uses a different type of neuron rather than the sigmoid one for feature extraction. This type of neuron is based on the *shunting inhibitory* mechanism, which has been used to model a number of visual and cognitive functions, see, e.g., [14]. Recently, shunting inhibitory neurons have been used for supervised classification and regression problems, where experiments have shown that shunting neurons are more powerful than their sigmoid counterparts [15].

The neural activity of a shunting neuron in the k^{th} feature map of the L^{th} layer centered at position (i, j) is given by

$$Z_{L,k}(i,j) = \frac{X_{L,k}(i,j)}{a_{L,k}(i,j) + Y_{L,k}(i,j)}, \tag{1}$$

where

$$X_{L,k}(i,j) = f_L\Big(\sum_{m=1}^{S_{L-1}} [C_{L,k} * Z_{L-1,m}]_{(x,y)} + b_{L,k}(i,j)\Big)$$

$$Y_{L,k}(i,j) = g_L\Big(\sum_{m=1}^{S_{L-1}} [D_{L,k} * Z_{L-1,m}]_{(x,y)} + d_{L,k}(i,j)\Big)$$

$$(x,y) = \begin{cases} (2i, 2j), & \text{First hidden layer (L=1);} \\ (i,j), & \text{Second hidden layer (L=2).} \end{cases}$$

$$\forall \;\; i,j = 1, ..., M_L$$

The weights of the receptive fields of the neuron, $C_{L,k}$ and $D_{L,k}$, are adaptive, as well as the biases $b_{L,k}$ and $d_{L,k}$, and the passive decay term $a_{L,k}$. M_L is the size of the feature map. After some preliminary experimentations, both activation functions f_L and g_L in the first hidden layer are chosen as linear functions, except that g_L is bounded from below by zero:

$$f(x) = max(0, x). \tag{2}$$

However, in the second hidden layer, f_L is chosen as the hyperbolic tangent function. In order to avoid division by zero in (1), the passive decay rate is constrained during both the initialization and training process as follows:

$$[a_{L,k}(i,j) + Y_{L,k}(i,j)] \geq \varepsilon > 0, \tag{3}$$

where $\varepsilon = 0.1$. To classify the features extracted from the second hidden layer, sigmoid type neurons are used in the output layer. Each output neuron performs a weighted sum and a nonlinearity operation on its input signals. That is, the response of the output neuron is computed as

$$y = h\Big(\sum_{i=1}^{S_N} w_i z_i + b\Big), \tag{4}$$

where h is the hyperbolic tangent activation function, w_i's are the connection weights, z_i's are the net signals to the neuron, which are averaged features from the second hidden layer, S_N is the number of inputs, and b is the bias term.

The connection strategy used in the network developed by Jain and Karu [9] is a full connection scheme, which subsequently adapted to a partial connection scheme with a pruning algorithm. In contrast, the CoNN proposed here has a binary connection scheme, in which each feature map branches to two feature maps in the next layer, similar to a binary tree. Moreover, depending on the complexity of the input, the number of feature maps in the first hidden layer (F1) can be chosen arbitrarily, and the subsequent hidden layer (F2) is constrained to have twice the number of feature maps. In other words, the feature map in F1 extracts local features from the input image; each feature is further decomposed into two other features. A schematic diagram of the proposed CoNN for texture classification is shown in Fig 1.

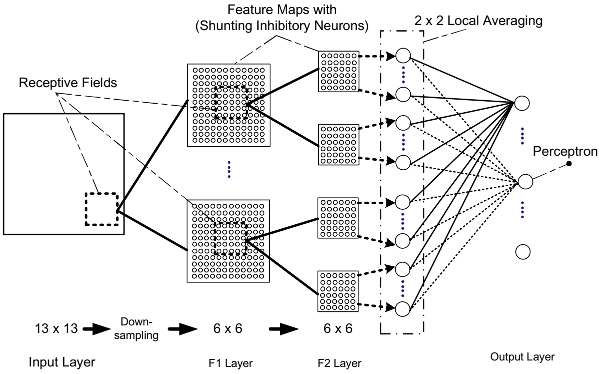
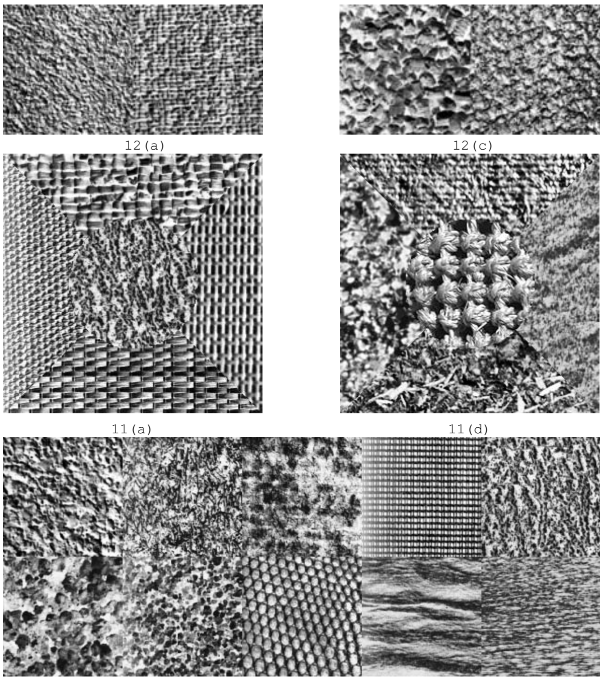


Fig. 1. A schematic diagram of the binary-connected CoNN for texture classification.

III. Training and Evaluation Procedures

Before starting the training process of the network, its weights are initialized with random values using a uniform distribution on the interval $[-1/t, 1/t]$, where t is the width of the receptive field. The bias parameters are initialized similarly with $t = 1$. However, due to the condition given in (3), the passive decay rate term is initialized in the range $(0, 1]$. The weight adjustment is done with a first-order gradient method derived from the training algorithms Rprop, QuickProp and SuperSAB (see [12] for more details), in which each weight (including biases and passive decay rate) has its own learning parameters (i.e., learning and momentum rates). Since it is a batch training, the weight update and the computation of the gradient are performed after presenting all input patterns.

The Brodatz texture database has become the standard data set for evaluating texture algorithms which is derived from the Brodatz album [16]. Randen and Husøy [2] created a set of texture mosaics from these Brodatz images to compare different classification approaches including the neural technique developed by Jain and Karu. To evaluate the performance of the proposed CoNN, five test texture mosaics, i.e., 11(a), 11(d), 11(h), 12(a) and 12(c), have been used; these images are available from the website [17]. The texture mosaics used in our experiment vary in terms of number of textures and granularity, as shown in Fig. 2. The grayscale texture mosaics are linearly scaled to the range $[-1, 1]$ before training and testing, and the target value is 1 for the correct class and -1 otherwise.



12(a)

12(c)

11(a)

11(d)

11(h)

Fig. 2. Texture images used to evaluate the proposed network.

IV. Experimental Results and Performance Analysis

As the texture mosaics have different number of textures to be classified, three different networks were developed; their network configurations are given in Table I. The sizes of training sets vary from 13000 to 19000 samples according to the number of textures, and all networks were trained for 1000 iterations. From Table I, all three CoNNs have fewer trainable parameters than the neural network developed by Jain and Karu [9]. The classification errors of these networks based on the five texture mosaics are given in Table II, together with the classification results of other approaches, such as the co-occurrence matrix, Gabor, wavelet, and QMF filters, and Jain's neural network; the results in Table II are taken from Tables 3 and 6 in [2].

TABLE I

Network configuration used for different number of texture classes.

Network Index	No. of weights	No. of feature maps		No. of output neurons
		Layer 1	Layer 2	
Net-01	974	4	8	2
Net-02	1490	5	10	5
Net-03	3106	8	16	10

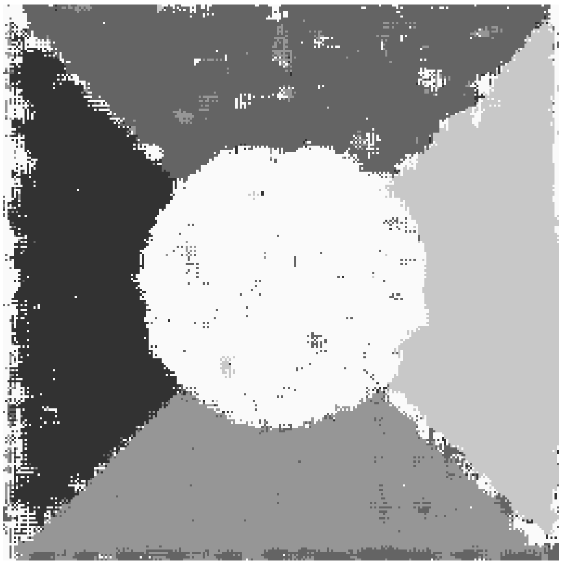
TABLE II

Error rates of different texture classification approaches.

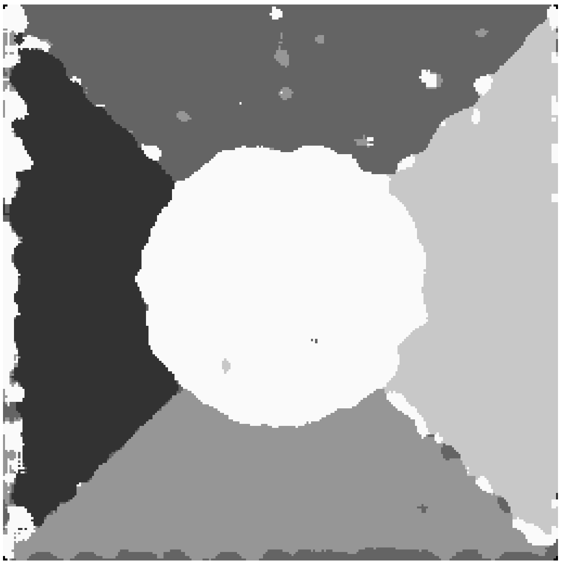
Texture classification Approach	Test image from [17]					Mean
	11(a)	11(d)	11(h)	12(a)	12(c)	
CoNN mask size 13	9.7	25.6	37.8	3.4	9.4	17.2
CoNN + Median filter	7.5	18.3	32.3	1.6	7.2	13.4
Neural Net mask size 11 [9]	47.4	67.2	69.3	32.1	43.6	51.9
Co-occurrence	9.9	51.1	35.3	1.9	3.3	20.3
Gabor filter bank	8.2	36.9	39.7	6.5	15.6	21.4
Wavelet - Daubechies 4	8.7	23.4	40.9	5.7	8.2	17.4
QMF filter bank - f16b (d)	8.7	18.4	39.8	8.1	8.2	16.6

From Table II, it can be concluded that our CoNN can be employed for texture classification; its classification accuracy is slightly better than or comparable to that of the best existing approaches, i.e., wavelets and QMF filter banks. When using a 5 × 5 median filter on the final outputs of the network, the classification errors across all five texture images decreases markedly. On the two texture images with complex border, the CoNN outperforms the co-occurrence and Gabor filter methods. Figure 3 shows an example of the output texture image with and without median filtering.

Even though the CoNN structure uses similar concepts taken from LeCun *et al.* [13], i.e., weight sharing and subsampling, the network implementation and its connection scheme are different, which yields a smaller network with fewer trainable parameters. For example, Jain and Karu [9] developed a network that has around 5000 weights to classify nine textures, whereas our CoNN has 3106 weights for ten textures, i.e., a reduction of 1894 weights.



(a)



(b)

Fig. 3. Segmentation of the texture image 11(a): (a) output image from the CoNN, and (b) post-processed by a median filter.

V. Conclusion

Many texture classification algorithms have been reported in the literature including neural networks. However, the existing neural architectures are too massive in terms of number of trainable parameters, and hence may require a pruning method. In this paper, we developed a CoNN that can be trained with a batch training algorithm to adapt its receptive fields as texture filters and its output layer as a linear classifier, where both processes are performed within the same network architecture. Evaluated on some benchmark texture images, the CoNN achieves superior classification results than the existing neural networks and has very competitive results with respect to the wavelets and QMF filter banks. Postprocessing the outputs of the CoNN with median filtering significantly improves the classification accuracy.

References

[1] M. Tuceryan and A. K. Jain, "Texture analysis," in *The handbook of pattern recognition and computer vision*, 2nd ed., C. Chen, L. F. Pau, and P. Wang, Eds. Singapore: World Scientific, 1998, pp. 207–248.

[2] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, 1999.

[3] R. M. Haralick, "Statistical and structural approaches to texture," in *Proc. IEEE*, vol. 67, 1979, pp. 786–804.

[4] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25–39, 1983.

[5] X. Liu and D. Wang, "Texture classification using spectral histograms," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 661–670, 2003.

[6] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[7] T. Randen and J. H. Husøy, "Multichannel filtering for image texture segmentation," *Optical Engineering*, vol. 33, pp. 2617–2625, 1994.

[8] B. W. Ng and A. Bouzerdoum, "Supervised texture segmentation using DWT and a modified K-NN classifier," in *Proc. of the 15th International Conference on Pattern Recognition*, vol. 2, 2000, pp. 545 – 548.

[9] A. K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 195–205, 1996.

[10] Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communication*, pp. 41–46, 1989.

[11] J. Mao, K. Mohiuddin, and A. K. Jain, "Minimal network design and feature selection through node pruning," in *Proc. of the 12th International Conference on Pattern Recognition*, vol. 2, 1994, pp. 622–624.

[12] F. H. C. Tivive and A. Bouzerdoum, "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 541–556, 2005.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] S. Grossberg, Ed., *Neural Networks and Natural Intelligence*. Cambridge, Massachusetts: MIT Press, 1988.

[15] G. Arulampalam and A. Bouzerdoum, "A generalized feedforward neural network architecture for classification and regression," *Neural Networks*, vol. 16, no. 5-6, pp. 561–568, 2003.

[16] P. Brodatz, "Texture: A photographic album for artists and designers," 1966.

[17] T. Randen. Personal homepage. Available: http://www.ux.his.no/ tranden/data.html.