Derong Liu   Shumin Fei
Zengguang Hou   Huaguang Zhang
Changyin Sun (Eds.)

Advances in Neural Networks – ISNN 2007

4th International Symposium
on Neural Networks, ISNN 2007
Nanjing, China, June 3-7, 2007
Proceedings, Part III

Springer

Image Processing Applications with a PCNN

Mario I. Chacon M., Alejandro Zimmerman S., and Pablo Rivas P.

DSP & Vision Lab, Chihuahua Institute of Technology, Mexico
mchacon@itchihuahua.edu.mx,
pablorp80@ieee.org

Abstract. This paper illustrates the potentials of the PCNN for image processing. A description of three schemes for image processing using the PCNN is presented in this paper. The first scheme is related to image segmentation, the second to automatic target location, ATL, and the third to face recognition. The first scheme was developed in order to obtain an insight of the behavior of the PCNN as a preprocessor element, the second one is an application to test the performance of the PCNN in an ATL problem. The third is a feature extraction method for face recogiton. The segmentation scheme showed great potentials to perform pixel grouping. The second scheme turned into a system with an ATL performance as good as other systems reported in the literature. And the third scheme seems to improve the performance of a face recognition system.

1 Introduction

The Pulse Coupled Neural Network, PCNN, is a relative new ANN model with a great potential in the area of image processing. A PCNN, is a model derived from a neural mammal model [1]-[4]. Current works with PCNN document how the PCNN can be used to perform important image processing task; edge detection, segmentation, feature extraction, and image filtering [2]-[10]. Because of this kind of performance the PCNN is considered a good preprocessing element.

The basic model of a neuron element of a PCNN has three main modules: the dendrite tree, the linking and the pulse generator [1]. The dendrite tree includes two special regions of the neuron element, the linking and the feeding. Neighborhood information is incorporated through the linking. The input signal information is obtained through the feeding. The pulse generator module compares the internal activity, linking plus feeding activity, with a dynamic threshold to decide if the neuron element fires or not. Fig.1 illustrates the basic model of the PCNN. A PCNN mathematical definition is given by (1) to (5). Equation (1) corresponds to the feeding region of the neural element, where G_{Feed} is the feed gain, S is the input image, $\alpha_{F\Delta t}$ is the time constant of the leakage filter of the feeding region, $Y(t)$ is the neuron output at time *t,* and W is the feeding kernel. The outputs $Y(t)$ of the PCNN can be observed as output images called pulsed images of the PCNN. Equation (2) describes the linking activity. Here G_{Link} is the linking gain, $\alpha_{L\Delta t}$ is the time constant of the leakage filter of the linking region, and M is the linking kernel. Equation (3) corresponds to the

D. Liu et al. (Eds.): ISNN 2007, Part III, LNCS 4493, pp. 884–893, 2007.

internal activity of the neuron element. The internal activity depends on the linking and feeding activity. In (3) β is the linking coefficient. β defines the amount of modulation of the feeding due to the linking activity. The dynamic threshold is implemented by (4), where α_θ is the time constant of the leakage filter of the threshold and V is the threshold gain. Finally the output of the neuron is defined by (5). In the case of an image processing task, each pixel is related to a neural element. For more information on how a PCNN works consult [1].

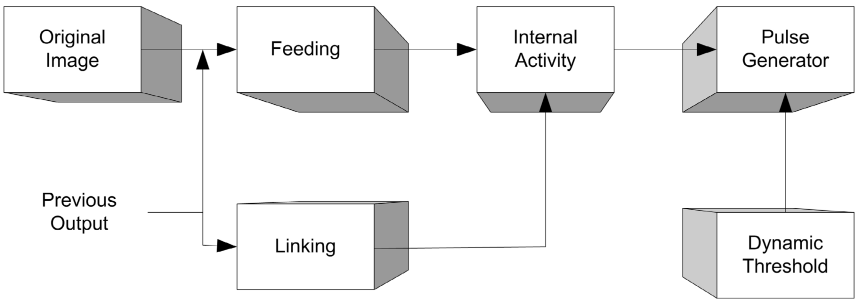

Fig. 1. Basic model of the PCNN

$$F(t) = G_{Feed} e^{-\alpha_F \Delta t} F(t-1) + S + Y(t-1) * W \, , \tag{1}$$

$$L(t) = G_{Link} e^{-\alpha_L \Delta t} L(t-1) + Y(t-1) * M \, , \tag{2}$$

$$U(t) = F(t)\left[1 + \beta L(t)\right], \tag{3}$$

$$\theta(t) = e^{-\frac{1}{\alpha_\theta}} \theta(t-1) + VY(t), \tag{4}$$

$$Y(t) = \begin{cases} 1, & \text{if } \; U(t) \rangle \theta(t), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2 Segmentation Experiment with the PCNN

One important characteristic of the PCNN is the grouping property. Since the fire of a neuron element depends on the input information and the neighbor information, it is possible to make that a set of neurons fire at the same time. This situation occurs when the neurons are related to image pixels corresponding to a uniform region. Taking advantage of this property we use the PCNN to perform segmentation. Fig 2 illustrates the pulsed images of the PCNN. In this application a noisy image, composed of a black square and noise, was processed by a PCNN using low pass filters as the feeding and linking kernels and using the basic PCNN model. In this case low pass filter kernels are used since the purpose is to emphasize the ability to generate uniform regions, as observed in pulse 4. It can be observed in pulsation 4 that the object is practically defined.

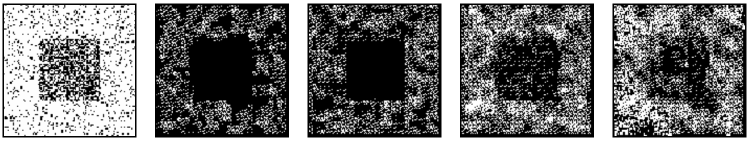

Fig. 2. Noisy image processed by the PCNN for segmentation purpose

3 PCNN Automatic License Plate Location Scheme

This scheme shows the application of the PCNN in automatic target location, ATL. The problem to solve is to locate license plates on digital images using a scheme based on a dynamic PCNN network. The images tested with the proposed ATL system are taken without special conditions, which is a difference with other works reported in the literature [11]-[13].Not special conditions means; images acquired with a video camera, no special illumination equipment, and there is not restriction about the position of the license plate. The purpose of not restrict the acquisition stage is to evaluate the PCNN system with similar conditions where a person will succeed.

The dynamic PCNN ATL model that we proposed is an iterative model based on the PCNN, Fig. 3. The original image is preprocessed by the PCNN. The PCNN pulses and generates an output image. The PCNN is designed to yield regions [6], such that they may contain candidate regions of the license plate. A whole license plate segmentation is guaranteed because of the region constitution of the license plate. These regions are then analyzed to obtain labeled regions, $r_n(x, y)$. These candidate regions are analyzed using area, and the minimum rectangle features to keep only regions that are good candidates, $r_{ng}(x, y)$. After this process, statistics of the Fourier transform of the candidate regions are computed. These statistics are then used to decide if the region contains or not a license plate. If any of the regions contains the license plate then the process iterates to the point where the PCNN pulses again. In this new iteration the parameters of the PCNN are redefined, making a dynamic process. The objective of the change of parameters is to force the PCNN to generate smaller regions than the ones generated in the previous step. In the first iteration the parameters are adjusted to generate big regions. The parameters are adjusted after the second iteration to yield smaller regions. The initial parameters of the PCNN and the updated parameters are shown in Table 1. The kernels *W* and *M* are 3x3 average kernels, because average kernels reinforce grouping. At the time of the realization of this work it was not known an analytic procedure to determine the parameter changes, thus the parameters were estimated by experimentation using a PCNN processor software [9], and based on the knowledge of the PCNN behavior.

Table 1. PCNN parameters

First iteration	Next iterations
$\beta = 1.0$	$\beta = 1.0$
$G_{Feed} = 0.2$	$G_{Feed} = 0.7$
$\alpha_F = -0.1$	$\alpha_F = -0.1$
$G_{Link} = 0.1$	$G_{Link} = 0.6$
$\alpha_L = -0.1$	$\alpha_L = -0.1$
$\alpha_\theta = 0.75$	$\alpha_\theta = 0.75$
$V = 10$	$V = 10$

A first discrimination process of regions is used to discard regions like point, lines, etc. Discrimination of regions is performed by (6).

$$f(A, M_r) : r_n(x, y) \rightarrow r_{ng}(x, y), \tag{6}$$

where A is the area of the region and M_r is the minimum rectangle. Equation (6) analyzes the possibility of a $r_n(x, y)$ to be a $r_{ng}(x, y)$ based on the features area and minimum rectangle. Each region $r_{ng}(x, y)$ is then represented by a feature vector consisting of statistics of the Fourier transform of the region as follows

$$R_{ng}(u, v) \rightarrow \mathbf{T} = \begin{bmatrix} \mu \\ \sigma \end{bmatrix}, \tag{7}$$

where μ, and σ, are the mean and standard deviation, of $R_{ng}(u, v)$. The representative vectors for each class are, region with plate, $\mathbf{T}_{wp}$, region without plate, $\mathbf{T}_{np}$. $\mathbf{T}_{wp}$ and $\mathbf{T}_{np}$ were determined by statistical analysis. Under this scheme if the condition $\| \mathbf{T}_{wp} - \mathbf{T}_i \| > \| \mathbf{T}_{np} - \mathbf{T}_i \|$ is true for all regions $R_{ngi}(u, v)$, then the pulsed image does not have a region with the license plate and the process iterates to the point where the PCNN pulses again. Fig 4 shows an example of a car image and the license plate found.

3.1 Results for License Plate Location

The segmentation scheme yield adequate results when the PCNN is used with low pass filter kernels.

High pass filters kernels were also implemented to achieve segmentation by edges; however, results were not satisfactory. In the case of the ATL scheme, the proposed system was tested with a database of 60 images acquired with a video camera without special conditions of illumination. Results yield an 85% of correct license plate location. Other systems to locate license plate report results from 80% to 95%, of correct location under special conditions [14]-[16]. Based on these results it can be said that the PCNN architecture may provide important advantages in the preprocessing stage of images.

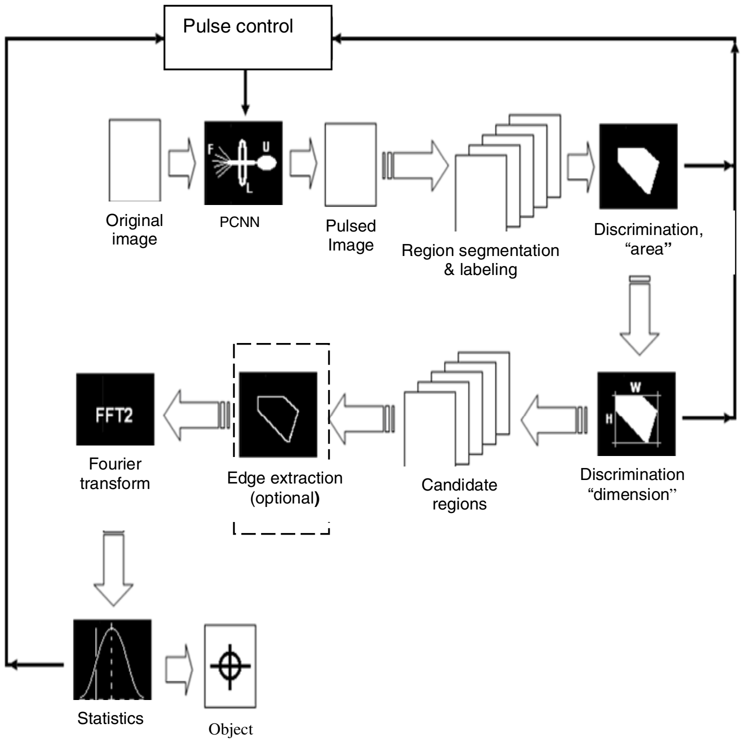

Fig. 3. Dynamic PCNN ATL model

Fig. 4. Original image and the plate located in the image

4 PCNN in the Face Recognition Problem

The problem related to face recognition has been investigated from different points of view. Artificial Neural Networks, ANN, is not the exception. In this section we present a novel approach for face recognition based on three features, first the Pulse Coupled Neural Network, PCNN; second, the Hough transform, HT; third, the Karhunen-Loeve transformation. The facial features are introduced in a neurofuzzy system based on the fuzzyfication of the inputs on an RBF neural network with a variable architecture on the first layer. The system performs well with ORL and YALE face databases, reaching recognition rates comparable with current face recognition systems.

The automatic face recognition systems have 5 main stages: Input, Motion Detection/Face Detection, Feature Extraction, Face Recognition, and the Output/Identification, as shown in Fig 5. Here we use the PCNN as feature extraction method for a face recognition system.

4.1 PCNN Initial Parameters

For this particular project, the initial parameters of the PCNN are shown in Table 2. The kernels W and M are 3x3 average kernels, because average kernels reinforce grouping.

4.2 PCNN Facial Feature Extraction

In previous sections we have defined what a pulsed image is, as a result of applying the PCNN to a given image. Now, if a gray-scale image is given to the input of the PCNN, we obtain pulsed images similar to the ones in Fig. 6 where it is shown that the pulsations of the faces changes across time. The selected pulses are the 36 to 40. These pulsations are selected because their content of facial information is useful to construct a feature vector. As in Fig. 6 the pulses 36 to 40 have more content rather than the first 10 pulsations.

The original image $I(x,y)$ is preprocessed by the PCNN. The PCNN pulses and generates an output image $I_p(x,y)$. The canonical form of a pulsed image $I_p(x,y)$ is $\mathbf{i}_{p_{40}}$ (row vector). The pulsed images, 36 to 40, from the PCNN are collected to generate a feature vector,

$$\mathbf{T}_{pcnn} = \begin{bmatrix} \mathbf{i}_{p_{36}} & \mathbf{i}_{p_{37}} & \dots & \mathbf{i}_{p_{40}} \end{bmatrix}. \tag{8}$$

The original image face is reduced to the static size of 20×18 pixels no matter what the original size is. This amount of reduction is performed only for this particular part of the feature vector. Now we have that the size of $\mathbf{T}_{pcnn}$ will be 1800 because $xy\times 5 = 20\times 18\times 5 = 1800$.

Table 2. PCNN initial parameters

For 50 iterations
$\beta = 1.0$
$G_{Feed} = 0.1$
$\alpha_F = 0.1$
$G_{Link} = 1$
$\alpha_L = 0.1$
$\alpha_\theta = 5$
$V = 5$

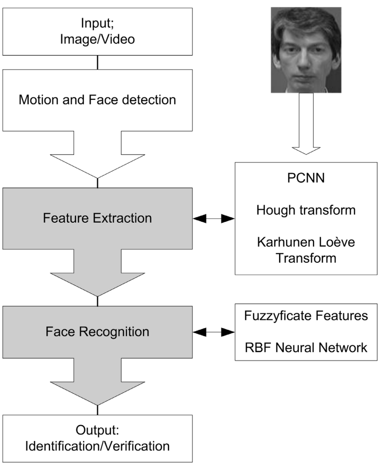

Fig. 5. Proposed face recognition system using PCNN as feature extractor

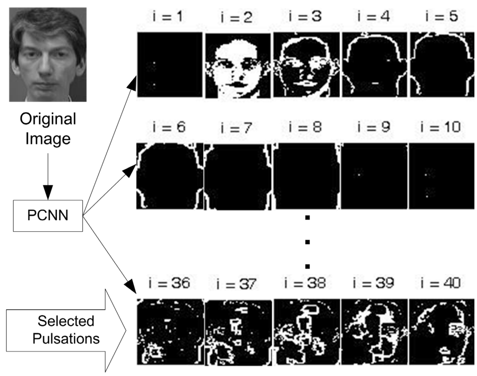

Fig. 6. Images pulsed by a PCNN when a face image is the input. Pulsations from 36 to 40 are selected as features.

4.3 Final Face Feature Vector

The final face feature vector includes three main components, face feature lines FFL $\mathbf{z}_i$, Karhunen-Loève feature transformation [14], KLT $\hat{\mathbf{i}}_{xy}$ (a variation of the PCA), and the features extracted with the PCNN $\mathbf{T}_{pcnn}$. Face feature lines are four prominent lines that can be extracted with the Hough transform from low resolution image faces, and are important features documented in newborn face recognition studies [15]. The KLT features are defined by

$$\hat{\mathbf{i}}_{xyn} = \mathbf{W}_{KLT}^{T}\mathbf{i}_{xyn}\,, \tag{9}$$

where $\mathbf{i}_{xyn}$ is the whole training set composed by n $\mathbf{i}_{xy}$ vectors, $\mathbf{i}_{xy}$ is the canonical form of the original $I(x,y)$, and $\mathbf{W}_{KLT}^{T}$ is the transformation matrix composed by the eigenvectors of the covariance matrix of $\mathbf{i}_{xyn}$.

The final feature vector can be now defined as

$$\mathbf{d}_{i+xy} = [\mathbf{z}_i \quad \hat{\mathbf{i}}_{xy} \quad \mathbf{T}_{pcnn}]\,. \tag{10}$$

4.4 Neuro-Fuzzy Network

The design of the network is based on a probabilistic RBF neural network with two layers and fuzzy inputs. The number of neurons on the first layer is the same as the number of the training samples. In this case we will be using different number of samples from 10 (one sample per class) to 80 (8 samples per class). Therefore we have from 10 to 80 neurons. The activation functions of the first layer are Gaussian. The number of neurons of the last layer is equal to the number of classes, in this work we will be recognizing ten people, therefore we have 10 classes, consequently 10 neurons. The way to fuzzificate the inputs of the network is achieved by membership functions for each component of the feature vector. These membership functions are created according to the distribution of each component for every single person. The input vectors $\mathbf{z}_i$ and $\hat{\mathbf{i}}_{xyn}$ are fuzzyfied with the membership functions just created.

4.5 Experiments and Results

We have experimented with two options to see the performance with and without PCNN. This experiment is one of the main contributions of this paper, to see how a PCNN improves or makes poor the performance of a face recognition system. The general scheme for the face recognition system when the PCNN is added to the system is shown in Fig 7. The experiments consist on changing the number of samples for training, selecting from 1 to 9 out of 10 samples per subject, and randomly selection of the samples. The experiments were designed to recognize 10 individuals. These experiments were realized over the OLR and YALE face databases. The testing results on the ORL without the PCNN have a maximum performance of 98%. For YALE without the PCNN the highest performance obtained reaches 78%. The testing results on ORL with PCNN show an improvement on 2 training samples (TS) for the subject #8 (S8) reaching 95%. The performance for YALE increases using the PCNN to 81%. Fig. 8 illustrates the performance of the experiments on the two databases, ORL and YALE with/without the PCNN. It is shown that the algorithm performs better on the ORL database because of less variation of the face samples regarding lighting conditions.

4.6 Conclusions of the Face Recognition Scheme

We have presented a neurofuzzy scheme for face recognition based on the PCNN feature extraction in a combination with the FFL and KLT features. The neurofuzzy algorithm was constructed extracting facial features via the PCNN, fuzzyfying the FFL and KLT features. These features are the inputs of an RBF neural network classifier which reaches 98% of recognition rate. This result is comparable to other systems previously developed [14]-[18]. As can be shown in Fig. 7 the algorithm performs well on ORL and it also performs better using PCNN over the YALE database even its severe lighting condition variations.

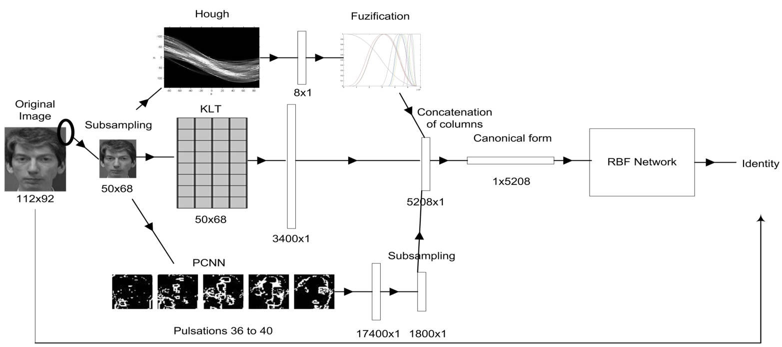

Fig. 7. General architecture for the PCNN neurofuzzy Hough-KLT face recognition system

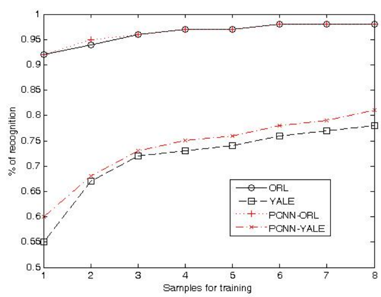

Fig. 8. Comparison of the experiments performed on the ORL and the YALE face databases with/without PCNN

5 Final Conclusions

This paper presented three applications of the PCNN architecture in image processing tasks. The first scheme is related to image segmentation, the second to automatic target location, and the third to face recognition. Results achieved in these three applications suggest that the PCNN architecture may be considered as a good preprocessor element to increase the performance of vision systems. The findings have demonstrated the potential of the PCNN to generate useful information especially in segmentation and feature extraction tasks. The two last applications have shown how the PCNN architecture can be incorporated in vision systems to achieve complex tasks.

Acknowledgment

The authors appreciate the support of COSNET, and SEP-DGEST for the support of this research under Grant 445.05-P. Also CONACyT under Grant #193324.

References

1. Lindbland, T., Kinser J.: Image Processing Using Pulse-Coupled Neural Networks. Springer (1998)
2. Johnson, J.L., Padgett, M.L.: PCNN Models and Applications. IEEE Transactions on Neural Networks **10** (3) (1999) 480-498
3. Johnson, J.: Pulse-Coupled Neural Nets: Translation, Rotation, Scale, Distortion, and Intensity Signal Invariance for Images. Applied Optics **30** (26) (1994) 6239-6253
4. Kuntimad, G., Ranganath, H.: Perfect Image Segmentation Using Pulse Coupled Neural Networks. IEEE Transactions on Neural Networks **10** (3) (1999) 591-598
5. Keller, P., Johnson, J.: Pulse-Coupled Neural Network for Medical Analysis. Proceedings of SPIE Applications and science of Computation Intelligence II (1999) 444- 451
6. Ranganath, H.S., Kuntimad G.: Object Detection Using Pulse Coupled Neural Networks. IEEE Transactions on Neural Networks **10** (3) (1999) 615-620
7. Rughooputh, H., Rughooputh S.: Spectral Recognition Using a Modified Eckhorn Neural Network Model. Image and Vision Computing (2000) 1101-1103
8. Ranganath, H.S., Kuntimad G.: Image Segmentation Using Pulse Coupled Neural Networks. IEEE World Congress on Computational Intelligence (1994) 1285-1290
9. Chacón, M.I., Zimmeman, A.: PCNNP: A Pulse-Coupled Neural Network Processor. Proceedings of the IEEE IJCNN (2002) 1581-1585
10. Chacón, M.I., Zimmeman A.: Image Processing Using the PCNN Time Matrix as a Selective Filter. Proceedings of the IEEE ICIP (2003) 1195-1199
11. Parisi, R., Di Claudio, E.D., Lucarelli, G., Orlandi, G.: Car Plate Recognition by Neural Networks and Image Processing. IEEE Proceedings of ISCAS-98 **3** (1998) 195-197
12. Yan, Y., Jin, G., Wu, M., He Q.: Optoelectronic Hybrid License Plate Recognition System. SPIE Conference on Applications of Digital Image Processing XXII **3808** (1999) 685-692
13. Zimic, N., Ficzko, J., Mraz, M., Virant J.: The Fuzzy Logic Approach to The Car Number Plate Locating Problem. IEEE Proceedings Intelligent Information Systes (1997) 227-230
14. Chacón, M.I.M., Rivas, P., Ramirez, G.: A Fuzzy Clustering Approach for Face Recognition Based on Face Feature Lines and Eigenvectors. IEEE International Seminar on Computational Intelligence (2006)
15. Rivas, P., Chacón, M.I.: Face Recognition Using Hough-KLT and a FeedForward Back-Propagation Neural Network. International Conference on Electronics Engineering, ELECTRO 2006 (2006)
16. Li, S.Z., Jain, A.K.: Handbook of face recognition. Springer (2004)
17. Li, W., Zhou, C.: Face recognition using feature Combination and Improved LDA. IEEE Transactions on Systems, Man, and Cybernetics **2** (2004) 1292-1296
18. Ekenel, H.K., Sankur, B.: Multiresolution Face Recognition. Image and Vision Computing **23** (2005) 469-477