Journal of Computer Science 6 (5): 506-510, 2010
ISSN 1549-3636

A Novel Approach to Classify Noises in Images Using Artificial Neural Network

[1]T. Santhanam and [2]S. Radhika
[1]Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India
[2] Department of Computer Applications, RMK Engineering College, Tiruvallur District, India

Abstract: Problem statement: Noise identification is predominant in any digital image processing algorithms, which helps in identifying the filters to smooth the image for further processing. **Approach:** An Artificial Neural Network (ANN) based approach was proposed for noise identification. The suggested technique involved seclusion of the noise samples and extracts their statistical features, which was then applied to a neural network to identify the noise. **Results:** Neural networks provided a better solution in identifying the noise. **Conclusion:** By identifying the noise the precise filter can be used for enhancing the given image.

Key words: Image noise, statistical parameters, ANN

INTRODUCTION

Image acquisition plays an essential role in image processing applications. However, state-of-the-art digital cameras are used to capture the images; there is always scope for image enhancement due to the presence of noise. Having identified the noise, the appropriate filter can be used to enhance the quality of the given digital image. In this article, an innovative approach for noise identification using a Neural Network is attempted.

An image is denoted by a two dimensional function of the form f(x,y). The function f(x,y) may be characterized by two components: (1) the amount of illumination on the scene being viewed and (2) the amount of illumination reflected by the objects in the scene. These components are the illumination and reflectance components and are denoted by i(x,y) and r(x,y), respectively. In other words, f(x,y) = i(x,y)r(x,y) (Gonzalez and Woods, 2002).

There are different types of image noise, viz. Gaussian noise, salt and pepper noise, speckle noise, shot noise and quantization noise. Understanding the noise characteristics helps in identifying the type of noise present in an image. A Neural network proves to be a good solution for the identification of noise by estimating the statistical features of the noise like skewness and kurtosis. Identification of the noise is crucial for an image denoising model to be accurate. Many image denoising algorithms were prevalent in the past, which can be applied to automatic image restoration (Luo, 2006; Figueiredo and Nowak, 2003; Chen *et al*., 1999; Buades *et al*., 2005; Portilla *et al*., 2003).

Identification of noise: In the literature, there are several techniques available to find the nature of the noise. The identification criteria used are based on the analysis of local variations of the average and the standard deviation of the observed image (Chehdi and Sabri, 1992). These statistical parameters are determined uniquely from the homogeneous regions.

Another approach consists of characterizing each class of noise by a parameter obtained from histograms computed on several homogeneous regions of the observed image. The homogeneous regions are obtained by segmenting images. Then, the estimation of the standard deviation is achieved from the analysis of a histogram of local standard deviations computed on each of the homogeneous regions (Beaurepaire *et al*., 1997).

Vozel *et al*. (2006) has proposed an unsupervised variational classification through a multi thresholding method. Chen and Das (2007) have proposed a simple pattern classification based noise identification by making use of the statistical features.

Central moments: Mean is the standardized first central moment of the probability distribution:

$$\mu = \Sigma x_i / n \quad (1)$$

where, x_i is the ith sampled value of the variable X.

Corresponding Author: T. Santhanam, Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India

Standard deviation is the standardized second central moment of the probability distribution:

$$\sigma = \sqrt{}\Sigma(x_i\text{-}x)^2/(n\text{-}1) \quad (2)$$

Skewness is the standardized third central moment of the probability distribution. Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point:

$$S = E(X\text{-}\mu)^3/\sigma^3 \quad (3)$$

Kurtosis is the standardized fourth central moment of the probability distribution. Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. That is, data sets with high kurtosis tend to have a distinct peak near the mean, decline rather rapidly and have heavy tails. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. A uniform distribution would be the extreme case:

$$S = E(X\text{-}\mu)^4/\sigma^4\text{-}3 \quad (4)$$

where the (-3) term makes the value zero for a normal distribution.

Skewness and Kurtosis are the statistical parameters used to identify the nature of the noise and they are measured as given in Eq. 3 and 4. Skewness is the opposite of symmetry. i.e., pixels that is darker or lighter than the mean. Kurtosis is a measure of peakedness. It represents for a given variance, the over representation or under representation of frequencies in the middle of the range compared to normal distribution. Non Gaussian white noise, Gaussian White noise and salt and pepper noise have been identified using this approach.

MATERIALS AND METHODS

Artificial neural networks: An Artificial Neural Network (ANN), usually called “Neural Network” (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Neural network models are essentially simple mathematical models defining a function f: X→Y. Each type of ANN model corresponds to a class of such functions. Neural network is popular because of its learning capability. Learning can be categorized as supervised, unsupervised and recurrent. The cost function C is an important concept in learning, as it is a measure of how far away we are from an optimal solution to the problem that we want to solve. Learning algorithms search through the solution space in order to find a function that has the smallest possible cost.

A Back Propagation Algorithm (BPN) follows supervised learning. In supervised learning, a set of example pairs (x, y), x ε X, y ε Y are given and the aim is to find a function f: X→Y in the allowed class of functions that matches the examples (Kumar, 2004). In other words, one wish to infer the mapping implied by the data; the cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

Back Propagation Neural Network (BPN): Back Propagation Neural Network is necessarily a multilayer neural network employing back propagation algorithm. Consider a neural network with n input and m output units. It can consist of any number of hidden units. When the input pattern x_i from the training set is presented to this network; it produces an output o_i different in general from the target t_i. o_i and t_i identical for i = 1, ..., p, by using the back propagation learning algorithm. To be precise the error is to be minimized:

$$E = \tfrac{1}{2}\sum_{i=1}^{p}\left\|o_i - t_i\right\|^2 \quad (5)$$

The steps in the BPN algorithm are (Kumar, 2004):

1. Select a pattern X_k from the training set T and present it to the network
2. Compute activations and signals of input, hidden and output neurons in that sequence
3. Find the error over the output neurons by comparing the generated outputs with the desired outputs
4. Use the error calculated in step 3 to compute the change in the hidden to output layer weights and the change in input to hidden layer weights (including all bias weights), such that a global error measure gets reduced
5. Update all weights of the network in accordance with the changes computed in step 4

Hidden to output layer weights:

$$W_{hj}^{k+1} = W_{hj}^{k} + \Delta W_{hj}^{k} \quad (6)$$

Input to hidden layer weights:

$$W_{ih}^{k+1} = W_{ih}^{k} + \Delta W_{ih}^{k} \quad (7)$$

where, ΔW_{hj}^{k} and ΔW_{ih}^{k} are weight changes computed in step 4.

Repeat steps 1-5 until the global error falls below a predefined threshold.

Multi Layer Perceptron (MLP): Multilayer perceptrons are feed-forward neural networks. They are supervised networks so they require a desired response to be trained. They learn how to transform the input data into a desired response and hence are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map. They have proved to approximate the performance of optimal statistical classifiers in difficult problems. Most neural network applications involve MLPs. The MLP architecture is shown in Fig. 1.

This network has an input layer (on the left) with three neurons, one hidden layer (in the middle) with three neurons and an output layer (on the right) with three neurons.

There is one neuron in the input layer for each predictor variable. In the case of categorical variables, N-1 neurons are used to represent the N categories of the variable.

Input layer: A vector of predictor variable values ($x_1 \ldots x_p$) is presented to the input layer. The input layer distributes the values to each of the neurons in the hidden layer.

Hidden layer: In the hidden layer, the value from each input neuron is multiplied by a weight (w_{ji}) and the resulting weighted values are added together producing a combined value u_j. The weighted sum (u_j) is fed into a transfer function, σ, which outputs a value h_j. The outputs from the hidden layer are distributed to the output layer.

Output layer: Arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight (w_{kj}) and the resulting weighted values are added together producing a combined value v_j. The weighted sum (v_j) is fed into a transfer function, σ, which outputs a value y_k. The y values are the outputs of the network.

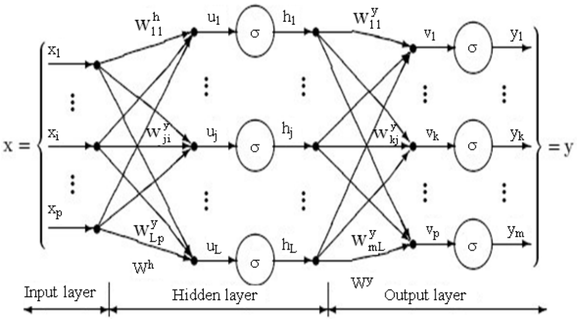

Fig. 1: Multilayer perceptron architecture

The goal of the training process in an MLP is to find the set of weight values that will cause the output from the neural network to match the actual target values as closely as possible. There are several issues involved in designing and training a multilayer perceptron network:

- Selecting how many hidden layers to use in the network
- Deciding how many neurons to use in each hidden layer
- Finding a globally optimal solution that avoids local minima
- Converging to an optimal solution in a reasonable period of time

Methodology: The suggested technique identifies the noise as non Gaussian white, Gaussian white and salt and pepper. These noises are additive in nature. The steps involved in the methodology used are summarized as follows:

- Get an image
- Introduce the noises (i.e., salt and pepper, non Gaussian white and Gaussian white)
- Get the noise samples by filtering salt and pepper noise using median filter and Gaussian noise using wiener filter
- Estimate the skewness and kurtosis of the noise samples
- At random, distribute the images to K partitions of equal sizes (say 10). Training is carried out with (K-1) partitions. (k-folding technique)
- Then, testing is performed with the left out partition
- The above process is repeated k times and during each cycle (ith cycle), ith partition is used for testing and the rest of the partitions are temporarily

combined for training the network and the resulting accuracy is recorded
- Lastly, the result of each run is added and final average is computed to determine the resulting accuracy

The CASIA-Irisv3 database has been used to test the proposed technique. (http://www.sinobimetrics.com) The database contains about 4500 iris images of which 250 left eyes and 250 right eye images were selected at random resulting in 500 images. Then, the three different types of noises were introduced with intensity as 0.02, making the mugshot database to 1500 images (i.e., 500 images with Non-Gaussian white noise, 500 images with Gaussian white noise and the rest 500 with salt and pepper noise). The experiment was performed using the MATLAB tool.

RESULTS

The confusion matrices given in Table 1 and 2 indicates the percentage of images classified correctly by the BPN and MLP network respectively for the given type of noise.

DISCUSSION

The BPN and MLP can be used to classify the noises to get the optimum accuracy. For Gaussian white and salt and pepper noise BPN yields an accuracy of around 90%, whereas for Non-Gaussian white MLP yields an accuracy of 88.66%. The experiments have been carried out in MATLAB and tested using the well known available CASIA-Irisv3 database. From Table 1 and 2, it is very evident that the performance of BPN is reasonably better than the MLP model. As a result, the appropriate filter can be used to reduce the noise, thus enhancing the given image for further processing.

Table 1: Confusion matrix-performance analysis of BPN

	Nongaussian white (%)	Gaussian white (%)	Salt and pepper (%)
Nongaussian white	81.80	13.1	5.10
Gaussian white	6.00	94.0	0.00
Salt and pepper noise	0.99	9.9	89.11

Table 2: Confusion matrix-performance analysis of MLP

	Nongaussian white (%)	Gaussian white (%)	Salt and pepper (%)
Nongaussian white	88.66	11.34	0.00
Gaussian white	6.59	93.41	0.00
Salt and pepper noise	3.57	13.39	83.04

CONCLUSION

The use of neural networks for noise identification using the statistical parameters is a novel attempt in this article. Neural Network models like BPN and MLP have been used to classify the noises and it’s apparent that the noises are classified to the optimum accuracy. As discussed the subsequent step is to identify the apt filter to reduce the noise in any given image thus making it suitable for processing.

REFERENCES

Beaurepaire, L., K. Chehdi and B. Vozel, 1997. Identification of the nature of noise and estimation of its statistical parameters by analysis of local histograms. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Apr. 21-24, IEEE Computer Society, USA., pp: 2805-2808. DOI: 10.1109/ICASSP.1997.595372

Buades, A., B. Coll and J.M. Morel, 2005. A non-local algorithm for image denoising. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 20-26, IEEE Computer Society, USA., pp: 60-65. DOI: 10.1109/CVPR.2005.38

Chehdi, K. and M. Sabri, 1992. A new approach to identify the nature of the noise affecting an image. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Mar. 23-26, IEEE Computer Society, San Francisco, CA., USA., 3: 285-288, DOI: 10.1109/ICASSP.1992.226195

Chen, T., K.K. Ma and L.H. Chen, 1999. Tri-state median filter for image denoising. IEEE Trans. Image Process., 8: 1834-1838. http://ieeexplore.ieee.org/iel5/83/17485/00806630.pdf

Chen, Y. and M. Das, 2007. An automated technique for image noise identification using a simple pattern classification approach. Proceedings of MWSCAS 07, IEEE Computer Society, USA., pp: 819-822. http://ieeexplore.ieee.org/iel5/4483493/4488527/04488699.pdf

Figueiredo, M.A.T. and R.D. Nowak, 2003. An EM algorithm for wavelet based image restoration. IEEE Trans. Image Process., 12: 906-916. DOI: 10.1109/TIP.2003.814255

Gonzalez, R. and R. Woods, 2002. Digital Image Processing. 3rd Edn., Prentice Hall Publications, USA., ISBN: 9788120336407, pp: 50-51.

Kumar, S., 2004. Neural Networks: A Classroom Approach. 1st Edn., Tata Mc-Graw Hill Publications, USA., ISBN: 0-07-048292-6, pp: 169.

Luo, W., 2006. Efficient removal of impulse noise from digital images. IEEE Trans. Consum. Elect., 52: 523-527. DOI: 10.1109/TCE.2006.1649674

Portilla, J., V. Strela, M.J. Wainwright and E.P. Simoncelli, 2003. Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Trans. Image Process., 12: 1338-1351. http://www.cns.nyu.edu/ftp/lcv/portilla03-preprint-corrected.pdf

Vozel, B., K. Chehdi, L. Klaine, V.V. Lukin and S.K. Abramov, 2006. Noise identification and estimation of its statistical parameters by using unsupervised variational classification. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, May 14-19, IEEE Xplore Press, Toulouse, pp: 841-844. DOI: 10.1109/ICASSP.2006.1660474