Image Noise Identification Using Neural Network with Convolution Matrix Technique

Ann Faik Sabeeh

Abstract:

The problem of noise identification in one-dimensional (signal) and two dimensional (image) processing is very important in engineering and computer science applications. These noises: salt, pepper, Gaussian and speckle, are different in subjective and objective behaviors. An Artificial Neural Network (ANN) based approach was proposed for noise identification. The suggested technique involved select the noise samples and extracts their statistical features using Convolution Matrix (CM), which was then applied to a neural network to identify the noise according to the distribution of the elements of the(CM) . Neural networks provided a better solution in identifying the noise.

Key words: Image noise, statistical parameters, ANN.

1 Introduction

In many engineering applications like astronomical, medical imaging, broadcast, and optical scanning, the observed images are subject to degradation caused by the electronic components or imaging environment. The degradation process model consists of two parts, the degradation function, and the noise function. If the observed image is the original image, it does not need a denoising [1]. Sometimes, we need to know the type of the noise that is appear in the noisy image depending on the application that we need. 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. The original image sometimes was found but in another times it does not found. Two types of algorithms are depending on it in this paper with and without original image using ***Convolution Matrix (CM)*** method.

The paper is organized as follows. Some background of the noise is briefly reviewed in section 2. Section 3 presents the ideas of the RM and DM, their generation, and their noisy images. Section 4 presents the DM. The proposed algorithm is introduced in Section 5. Experimental results are included in section 6.

2 The effect of noise in digital images The degradation process model consists of two parts, the degradation function, and the noise function. The general model in the spatial domain follows:

$$g(r,c) = h(r,c) \otimes\otimes x(r,c) + z(r,c) \quad \ldots(1)$$

Where: $\otimes\otimes$ denotes the two-dimensional convol-ution process, $g(r,c)$ represents theDegraded image, $h(r,c)$ represents the Degradation function, $x(r,c)$ represents the Original image, and $z(r,c)$ is the Additive noise function [6, 7].

This paper is concerning or dealing only with the noise degradation in the degradation process with the assumption that the degradation function $h(r,c)$ causes no degradation, so the only corruption to the image is caused by the noise (i.e. set $h(r,c) = 1$).

Noise is any undesired information that contaminates on image. The digital image acquisition process, which converts an optical image into a continuous electrical signal that is then sampled, is the primary process by which noise appears in digital images. At every step in the process there are fluctuations caused by natural phenomena that add a random value that exact brightness value for a given pixel. Another source of noise, which often arises from the electronic components in the image environment. Noise can be introduced into an image via a multitude of sources, such as communication channels, fault digital circuitry, and thermal noise. Film-grain type noise, which is data dependent, is caused when scanning an image recorded on photographic films [8, 9].

2.1 Noise Types [10]

In typical images, the noise can be modeled with a Gaussian "normal", uniform, speckle, or salt-and-pepper "impulsive" distribution.

2.1.1 Gaussian Noise

Gaussian noise takes the bell-shaped curve distribution, which can be analytically described by:

$$Histogram_{Gaussian} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(gl-m)^2}{2\sigma^2}} \quad \ldots(2)$$

Where **gl** is the gray level, **m** is the mean and σ is the standard deviation ($\sigma^2 = \text{variance}$)

About 70% of all the value full within the range from one standard deviation (σ) below the mean (m) to one above, and about 95% fall with in twos standard deviations. The Gaussian model is most often used to model natural noise process, such these occurring from electronic noise in the image acquisition system (*i.e. thermal noise*).

2.1.2 Speckle Noise

Speckle Noise is completely different from other types of noise, since it is operates on the images in multiplicative manner (i.e. the pixel value of the noise is multiplied by the pixel value of the original image), where the other types of noise mentioned above being additive (the pixel value of the noise is added to the pixel value of the

original image). The Speckle noise can be modeled by the following equation [1]:

$$g(r,c) = x(r,c) + x(r,c) \cdot * z_u(r,c) \quad \ldots(3)$$

Where $x(r,c)$, $z_u(r,c)$, and $g(r,c)$, are the original image, uniform noise (discussed in the previous section), and the distorted version image respectively. The operation $(\cdot *)$ is pixel-by-pixel multiplication. Figure 1 shows example of these types of noise, and how they are distributed.

2.1.3 Salt and Pepper Noise

In the salt-and-pepper noise model there are only two possible values, a and b, and the probability of each is typically less than 0.1. With numbers greater than 0.1 values, the noise will dominate the image. For an 8-bit image, the typical value for pepper-noise is 0 and for salt-noise 255.

$$Histogram \quad (S\&P) = \begin{cases} A & for \quad g = a \\ B & for \quad g = b \end{cases} \quad \ldots(4)$$

Malfunctioning pixel elements in the camera sensors, faulty memory locations, or timing errors typically causes the salt-and-pepper type noise. Usually, the speckle noise image appears similar to that of Gaussian noise images but darker. Salt-and-pepper type noise appears in the image as dots contaminate the image.

3 The Proposed Convolution Matrix(CM)

The Convolution Matrix (CM) is an (n×n-1) matrix where n is the number of gray scale levels of the image. Its dimension equal to the maximum levels in the image. The first pixel in the CM comes from the first pixel from the original matrix minus the second pixel in the original matrix in a row. These pixels are overlapped until reaching the end of the first row. This operation is repeated until all the rows of the original matrix are completed.

The CM is needed because it is used to recognize types of noise in the images like Gaussian and speckle noise. Using of the CM needs an algorithm to recognize these noises. The following steps of the CM algorithm are used. The CM structure is shown in Fig. (2).

4 Noise identification with CM

The following algorithm is used to identify each type of noise using original image:

1. Take the number of pixels that equal (0) (black) before and after adding the noise (name it **num1** and **num2**).
2. If the num1 equal num2 then we find a speckle noise in the image.
3. Find the difference between the absolute value of the number of one's before adding a noise and the absolute value of the number of one's after adding a noise.
4. If the difference is between a range taken then this indicate a salt and pepper noise.
5. If the conditions in steps 2 and 4 are not true then there is a Gaussian noise in the image.

5 Artificial neural networks

An Artificial NeuralNetwork (ANN), usually called "Neural Network" (NN), is a mathematical model or computational modelthat 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 tofind 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), xεX, y εY 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 bythe 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.

5.1 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 xi from the training set is presented to this network; it produces an output oi different in general from the target ti. oi and ti identical for i = 1, ..., p, by using the back propagation learning algorithm. To be precise the error is to be minimized:

$$E = 1/2 \sum_{i=1}^{p} \|o_i - t_i\|^2 \ldots (5)$$

The steps in the BPN algorithm are [9]:

1. Select a pattern Xk from the training set Tandpresent 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 output.

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:

$$Whj\ k+1 = (Whj\ k-1)+(Whj\ k)$$

Input to hidden layer weights:

$$Wih\ k+1 = (Wih\ k-1) + (Wih\ k)$$

where, Whj k and Wih k are weight changes computed in step 4.

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

5.2Multi Layer Perception (MLP)

Multilayer perceptions 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. (3). 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 ofthe variable.

Input layer: A vector of predictor variable values(x1...xp) 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 (wji) and the resulting weighted values are added together producing a combined value uj. The weighted sum (uj) is fed into a transfer function, σ , which outputs a value hj. 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 (wkj) and the resulting weighted values are added together producing a combined value vj. The weighted sum (vj) is fed into a transfer function, σ ,which outputs a value yk. The y values are the outputs of the network.

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 multilayerperceptron 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

6. 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:

1. Get an image.

2. introduce the noises (i.e., salt and pepper, non Gaussian white and Gaussian white).

3.Get the noise samples by filtering salt and pepper noise using median filter and Gaussian noise using wiener filter.

4.Add The CM to the noise samples.

5.At random, distribute the images to K partitions of equal sizes (say 10). Training is carried out with(K-1) partitions. (k-folding technique).

6.Then, testing is performed with the left out partition

7.The above process is repeated k times and during each cycle (i th cycle), i th partition is used for testing and the rest of the partitions are temporarily combined for training the network and the resulting accuracy is recorded.

8.Lastly, the result of each run is added and final average is computed to determine the resulting accuracy.

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

7. 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 ofaround 85%, whereas for Non-Gaussian white ML Pyields an accuracy of 82%. The experiments have been carried out in MATLAB and tested using the well known available. From Table 1and 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.

8. Experimental Results

An image (128×128) is used with other (99) images. The CM of the images, their noisy versions with different noise levels (1, 2, 3, … SNR). These (100) images are tested to the identification noise algorithms with NN. The result of identification was (98%). Fig. (4) gives an example. These algorithms are generated for each image Table (3,4) represents examples for these algorithms with the images in Fig.5.

9. Conclusions

In this paper a methods for identifying the noises was introduced. Its generation is very fast and can be generated for each image. The assumption for this method in that the images are smooth (grayscale levels of the image changes gradually). We found that this method does not give accurate results if the images have very sharp grayscale levels changes (non-smooth images).

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 ML P have been used to classify the noises and it's apparent that the noises are classified to the optimum accuracy.

10. References

1. Öktem R., 2000, " Transform Domain Algorithms for image denoisng and Compression", Tampere University of Technology, Finland.
2. Graps A., 1995, "An Introduction to Wavelet", IEEE Computational Science And Engineering, Vol.2, No.2, pp.1-19.
3. Lee J., 2000, " Noise Reduction in the Wavelet Domain", EE386 Digital Image Processing Class Project.
4. Waiel A., 1999, "Image Recognition Using Wavelet Transform", M.Sc. thesis, University of Baghdad, Elect. Eng. Dept.
5. Burrus G. S., Gopinath R. A., Guo H. , 1998, "Intriduction to Wavelet and Wavelet Transforms", Comm. Pure Appl. Math..
6. Umbaugh S. E., 1998, "Computer Vision and Image Processing ", a practical Approach.
7. Crandall R., Projects in Scientific Computation, Springer-Verlag, New York, 1994, pp. 197-198, 211-212.
8. DaubechiesI., 1988, "Orthonormal Bases of Compactly Supported Wavelets," Comm. Pure Appl. Math., Vol 41, pp. 906-966.
9. Santhanam T., Radhika S.,"A novel approach to classify noises in images using artificial neural network", Journal of computer science 6(5):541-545, 2010, ISSN 1549-3636.

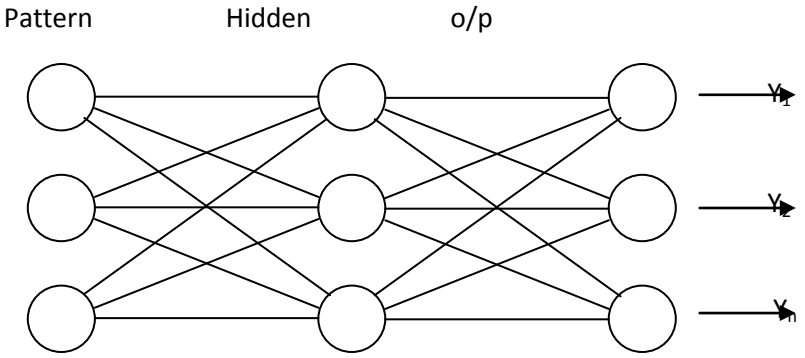

Figure (1): Back Propagation Neural Network

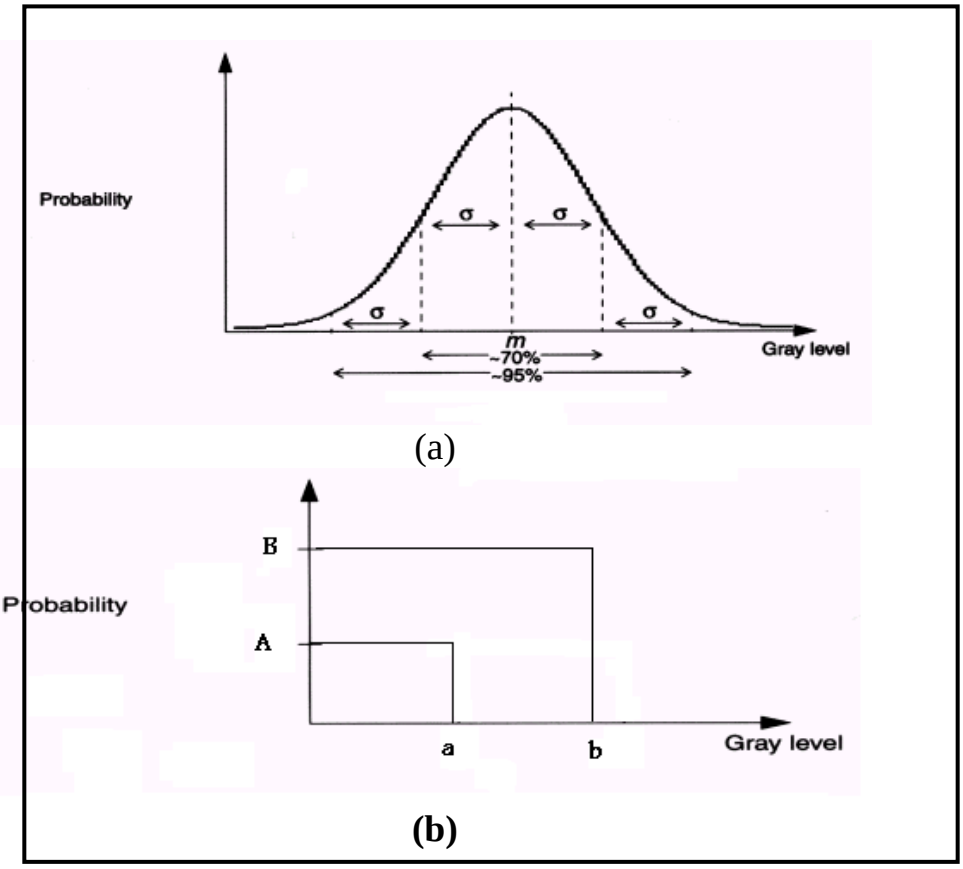

Figure (2): a) Gaussian distribution, b) Salt and pepper distribution.

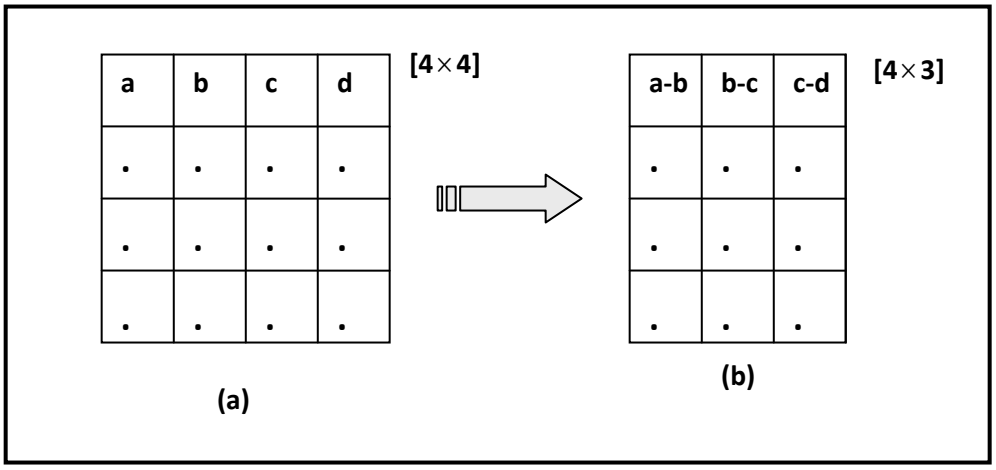

Figure (3): (a) Original matrix, (b) DM for matrix (a).

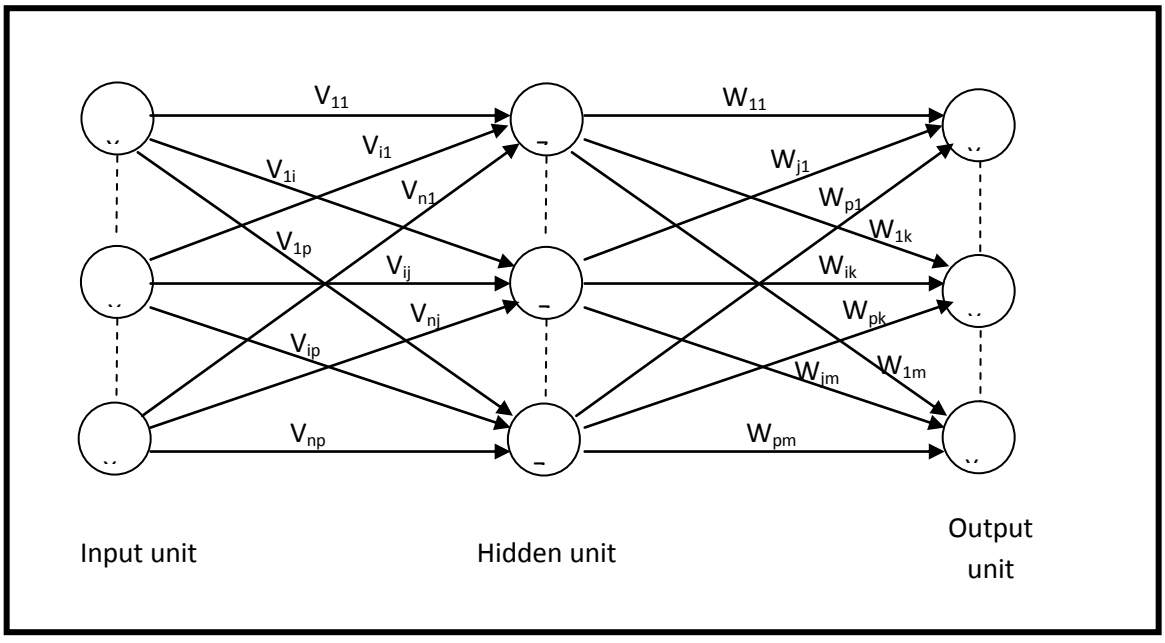

Figure (4):Multilayer perceptron architecture

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**

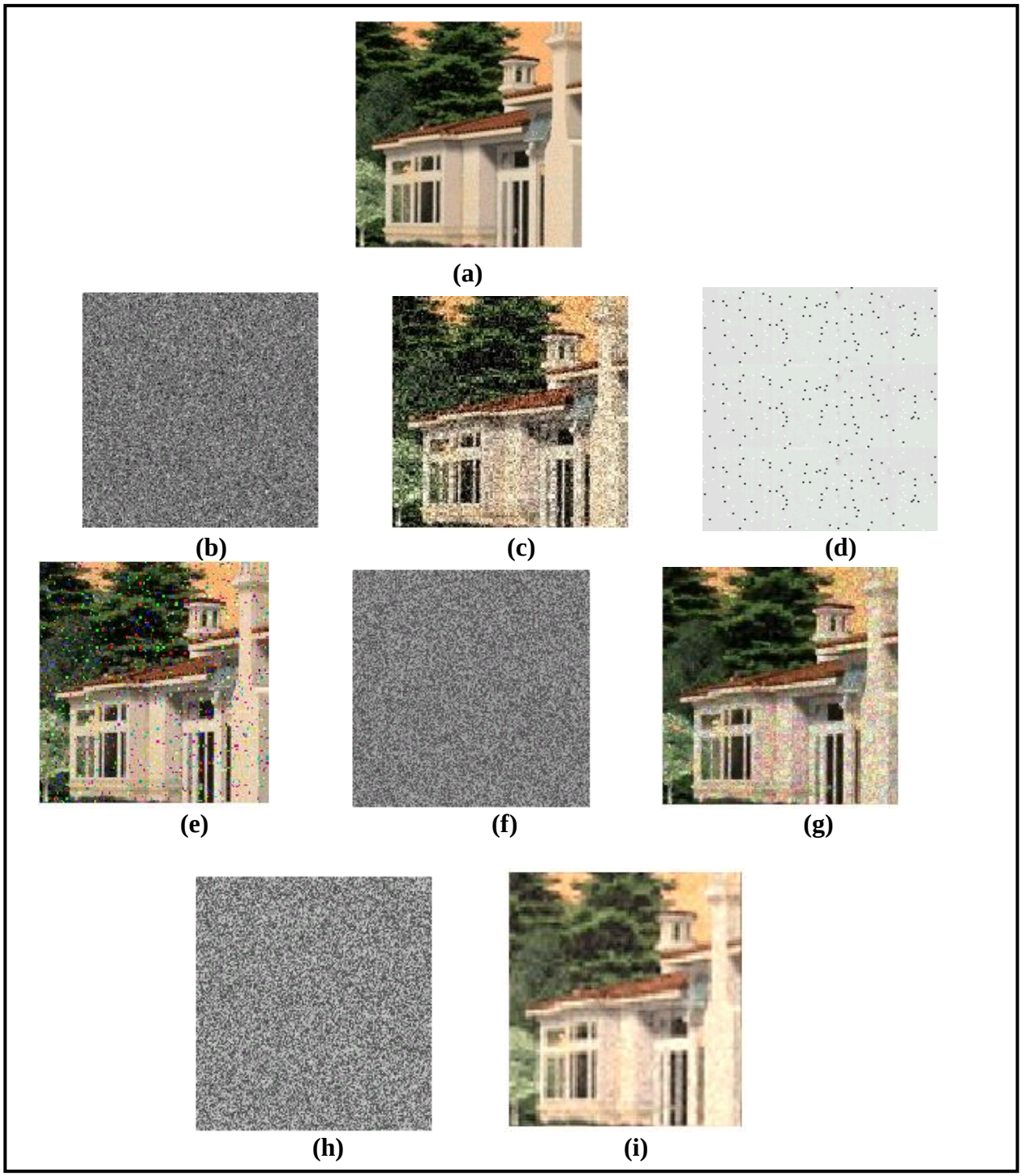

Figure (4): The effect of noise in several images: a) Original image, b) Gaussian noise– mean =0; σ =20, c) Original image with added Gaussian noise, d) salt and pepper noise with σ =20, e) Original image with

added salt and pepper noise f) Speckle noise - with variance = 0.04, mean = 0, g) Original image with added Speckle noise, h) Uniform noise, s= -50; t = +50, i) Original image with added Uniform noise.

Image name	speckle noise	S&P noise	Gaussian noise
I1	Speakle	S&P	**Speakle**
I2	**S&P**	**Speakle**	**Speakle**
I3	Speakle	S&P	Gaussian
I4	Speakle	S&P	Gaussian

Table (3): CA results with tested images.

Image name	speckle noise	S&P noise	Gaussian noise
I1	Speakle	S&P	Gaussian
I2	Speakle	S&P	**Speakle**
I3	Speakle	S&P	Gaussian
I4	Speakle	S&P	Gaussian

Table (4): NN with CA results with tested images.

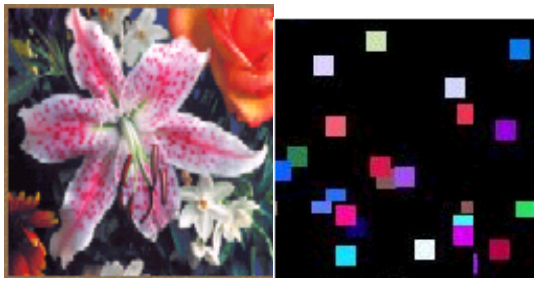

Figure (5): I1, I2, I3, I4 images.

التعرف على ضوضاء الصورة بأستخدام الشبكة العصبية

مع تقنية التواء المصفوفات

آن فائق صبيح

قسم علوم الحاسوب,كلية التربية أبن الهيثم, جامعة بغداد

الخلاصة:

أن مشكلة تحديد الضوضاء ذات البعد واحد (الأشارة) وذات البعدين (الصورة) مهم جدا خصوصا بالاختصاصات الهندسية وكذلك بالنسبة لتطبيقات علوم الحاسوب.
هذه الضوضاء من نوع (***salt, pepper, Gaussian and speckle***)تختلف من ناحية الموضوع والاهداف والسلوك. الشبكات العصبية أسست طريقة قد أقترحت لتحديد الضوضاء.
التقنية المقترحة قد تضمن أختيار عينات من الضوضاء وأستخراج الخصائص الاحصائية بأستخدام تقنية ألتواء المصفوفات التي سوف تطبق في الشبكات العصبية لتحديد الضوضاء أستنادا الى توزيع عناصر التواء المصفوفة, الشبكات العصبية توفر حل أفضل في تحديد الضوضاء.