

Cellular Neural Network as a Non-linear Filter of Impulse Noise

Elena Solovyeva

Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
selenab@hotmail.ru

Abstract—Feedforward discrete-time cellular neural network for filtering of impulse noise from two-dimensional (image) signals is represented. The parameters of mathematical filter model result from approximation problem solution in mean-square norm. It is shown that the cellular neural network surpasses median filter, Volterra filter and perceptron neural network in accuracy of image restoration and in simplicity of filter implementation.

I. INTRODUCTION

One of major problems in the communication theory is non-linear filtering of non-Gaussian noise. The impulse noise relates to non-Gaussian noise. This noise emerges during the switching of different electronic devices, in cases of mechanical damages of surfaces of the data storage devices, during the operation of internal-combustion engines, under the impact of various atmospheric phenomena, etc. Non-Gaussian noise distorts the working signals. Its emerging leads to the impairment of data transmission in communication channels [1–4].

Non-Gaussian noise is not cancelled by the methods of linear signal processing, so the methods of non-linear signal processing are used for non-Gaussian noise suppression, in particular, the methods within the framework of the "black box" principle. In view of this principle, the methods of building multidimensional polynomials and neural networks for non-linear devices, such as transformers, filters, compensators etc., are developed [5–7].

In recent years numerous non-linear filters are synthesized in the form of neural networks, which are universal approximators, and in some cases, they are simpler in comparison with multidimensional polynomials [6], [7].

The solving of an approximation problem results in establishing mapping between the input and output signals of a device and gaining high accuracy independently of computational cost (time, memory and etc.)

In this paper, the cellular neural network is considered as a model of non-linear filters for cancelling non-Cassian noise (for instance, impulse noise). The synthesis of combined filters based on the cascade connection of a median filter and neural network is proposed. The mathematical model of combined filter, containing the cellular neural network, is compared with the models of combined filters, including the perceptron

network and Volterra series. The numerical results of impulse noise filtering from distorted images show the advantage of using the cellular neural network over the perceptron network and Volterra series.

The rest of the paper is summarized as follows. In Section II, the problem for impulse noise filtering is introduced. Section III presents the concept of cellular neural networks. Section IV highlights the cellular neural network difference from the commonly used neural networks and the description of image signals and impulse noise. Section IV consists of numerical results of image restoring by various non-linear filters. The conclusions are drawn in the last section.

II. THE PROBLEM OF IMPULSE NOISE FILTERING FOR IMAGE RESTORATION

The problem of synthesizing the digital impulse noise filters can be solved within the framework of the "black box" principle [2], [5], [6], when filter operator F_S establishes a unique relationship between the set of input $s(n)$ ($s(n) \in S$) and output $y^O(n)$ ($y^O(n) \in Y^O$) signals of device

$$y^O(n) = F_S[s(n)].$$

For nonlinear dynamic system modeling it is required to approximate nonlinear operator F_S by nonlinear operator F_ε which reflects the input set X on the output set Y^O with error ε , $\varepsilon > 0$, that is

$$y(n) = F_\varepsilon[x(n)]$$

where $n \in [0, G_n]$ is the normalized discrete time, G_n is the duration of input signals.

The digital filter synthesis a consists in building a mathematical model in the form of operator equation

$$y(n) = F_\varepsilon[s(n)],$$

where $y(n)$ is the output signal of the filter model, F_ε is operator approximating F_S on sets S and Y^O under the condition

$$\|y^o(n) - y(n)\| \leq \varepsilon$$

for all $s(n) \in S$, $y^o(n) \in Y^o$. Here ε is the assigned error of simulation.

The parameters of nonlinear operator F_ε are determined by solving the approximation problem

$$\|y^o(n) - F_\varepsilon[s(n)]\| \rightarrow \min_D, \quad (1)$$

where D is the parameter set of operator F_ε . In practice, the approximation error is usually estimated in the mean-square norm.

The synthesis of combined filters (with an internal smoothing median filter (MF)) for image restoration [8]–[10], and application of neural networks as approximators [10] stipulate combined neural filter structure shown in Fig. 1. Here the block MF designates smoothing median filter, the block NN denotes neural network.

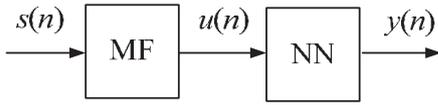


Fig. 1. The structure of combined neural filter

The set of signals $u(n)$ is interpreted as a set of fragments having length m , each of which is the vector composed of samples of the input signal at the template area (aperture) moving along the image with the step of one cycle.

The neural network (the block NN depicted in Fig. 1) can be specified in the form of cellular neural network.

III. CELLULAR NEURAL NETWORK

The cellular neural network (CNN) was introduced by Chua and Yang (1988) [11], [12]. It is a recurrent nonlinear network in which neurons are locally connected, and dynamics is identical for each node. These neurons are commonly called cells. The connection with the cells outside the r -neighborhood is enabled by the propagation effects of network dynamics. The CNN dynamics is described by a set of differential equations [13]–[18].

Each cell of CNN has an input, an internal state and an output. Any one cell is connected only to its neighboring cells. Cell located in the position (i, j) of two-dimensional $M \times N$ area is denoted as C_{ij} , and its r -neighborhood N_{ij}^r is defined by

$$N_{ij}^r = \{C_{kl}, \max(|k-i|, |l-j|) \leq r, 1 \leq k \leq M; 1 \leq l \leq N\},$$

where the size of the neighborhood r is a positive integer number.

Set N_{ij}^r is sometimes referred as the $(2r+1) \times (2r+1)$ neighborhood. For the 3×3 neighborhood, r should be 1. Thus, the parameter r controls the connectivity of a cell, i.e. the number of active synapses that connects the cell with its immediate neighbors.

CNN is entirely characterized by the set of nonlinear differential equations associated with cells in network. The mathematical model for the state equation of a single cell C_{ij} is given by the following relation:

$$\frac{\partial x_{ij}(t)}{\partial t} = -x_{ij}(t) + \sum_{kl \in N_{ij}^r} A_{ij,kl} y_{kl}(t) + \sum_{kl \in N_{ij}^r} B_{ij,kl} u_{kl}(t) + I_{ij}, \quad (2)$$

where $x_{ij}(t)$ denotes the state of cell C_{ij} ; $y_{kl}(t)$ and $u_{kl}(t)$ denote the output and input of cells C_{kl} located in the sphere of influence with radius r , respectively; t is the continuous time; $A_{ij,kl}$ and $B_{ij,kl}$ are the feedback and feedforward templates, respectively; I_{ij} is the bias term.

In many applications, CNN is isotropic, that is space-invariant. Isotropic network is characterized by parameters in equation (1) which are fixed for the entire neural network. In the case of isotropic CNN, for example under $r=1$, the terms of state equation (1) are represented below.

• Contribution of the feedback synaptic weights $A_{ij,kl}$ to equation (2). In view of space-invariance, we can write

$$\begin{aligned} \sum_{kl \in N_{ij}^r} A_{ij,kl} y_{kl}(t) &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} A(i-k, j-l) y_{kl}(t) = \\ &= a_{-1,-1} y_{i-1,j-1} + a_{-1,0} y_{i-1,j} + a_{-1,1} y_{i-1,j+1} + \\ &\quad + a_{0,-1} y_{i,j-1} + a_{0,0} y_{i,j} + a_{0,1} y_{i,j+1} + \\ &\quad + a_{1,-1} y_{i+1,j-1} + a_{1,0} y_{i+1,j} + a_{1,1} y_{i+1,j+1} = \\ &= \sum_{k=-1}^1 \sum_{l=-1}^1 a_{k,l} y_{i+k,j+l} = \\ &= \begin{bmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{bmatrix} \otimes \begin{bmatrix} y_{i-1,j-1} & y_{i-1,j} & y_{i-1,j+1} \\ y_{i,j-1} & y_{i,j} & y_{i,j+1} \\ y_{i+1,j-1} & y_{i+1,j} & y_{i+1,j+1} \end{bmatrix} = \\ &= \mathbf{A} \otimes \mathbf{Y}_{ij}, \end{aligned} \quad (3)$$

where the matrix \mathbf{A} of the 3×3 dimension is called the feedback cloning template, the symbol \otimes denotes the summation of dot products (or the sign of vector product), henceforth called a template dot product. In discrete mathematics, this operation is called “spatial convolution.”

The 3×3 matrix \mathbf{Y}_{ij} in (2) can be obtained by moving an opaque mask with the 3×3 window to the position (i, j) of the $M \times N$ output image Y , henceforth called the output image at $C(i, j)$.

An element a_{kl} is called the center (respectively, surround) element, the weight or coefficient, of the feedback template \mathbf{A} , if and only if $(k, l) = (0, 0)$ (respectively, $(k, l) \neq (0, 0)$).

• Contribution of the input synaptic weights $B_{ij,kl}$ to equation (2). Following the above notes, we can write

$$\begin{aligned} \sum_{kl \in N_{ij}^r} B_{ij,kl} u_{kl}(t) &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} B(i-k, j-l) u_{kl}(t) = \\ &= \sum_{k=-1}^1 \sum_{l=-1}^1 b_{k,l} u_{i+k, j+l} = \\ &= \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} \otimes \begin{bmatrix} u_{i-1, j-1} & u_{i-1, j} & u_{i-1, j+1} \\ u_{i, j-1} & u_{i, j} & u_{i, j+1} \\ u_{i+1, j-1} & u_{i+1, j} & u_{i+1, j+1} \end{bmatrix} = \\ &= \mathbf{B} \otimes \mathbf{U}_{ij}, \end{aligned} \quad (4)$$

where the 3×3 matrix \mathbf{B} is called the feedforward or input cloning template, and \mathbf{U}_{ij} is the translated masked input image.

• Contribution of the threshold term to equation (2). In view of space-invariance, denote $I_{ij} = z$.

Using the above notations in (3), (4), space-invariant CNN is completely described by state equation

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \otimes \mathbf{Y}_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z. \quad (5)$$

The output signal of cell C_{ij} is given by the following equation

$$y_{ij}(t) = f(x_{ij}(t)), \quad (6)$$

where $y_{ij}(t)$ denotes the output value of cell C_{ij} , $f(\bullet)$ is the non-linear activation function which is usually specified as the unity gain piecewise linear saturation function described by expression

$$y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2} \left(|x_{ij}(t) + 1| - |x_{ij}(t) - 1| \right)$$

and shown in Fig. 2.

The properties of the piecewise linear saturation function shown in Fig. 2 are the following:

- an uniform gain for low and high input signal amplitudes,
- quickness of signal conversion,
- implementation simplicity in the analog and digital fields using operational amplifiers and VLSI technology respectively.

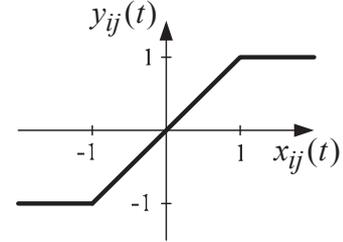


Fig. 2. The piecewise linear saturation function

A significant CNN feature is that CNN has two independent input capabilities: the generic input and the initial state of cells. They are normally bounded by $|u_{ij}(t)| \leq 1$ and $|x_{ij}(0)| \leq 1$. Similarly, if $|f(\cdot)| \leq 1$ then $|y_{ij}(t)| \leq 1$.

CNN is uniquely defined by three terms of the cloning templates $\{\mathbf{A}, \mathbf{B}, z\}$, which consist of 19 real numbers for the 3×3 neighborhood ($r = 1$).

One of the simplest CNN subclasses is zero-feedback (feedforward) CNN. CNN belongs to the zero-feedback subclass if and only if all the feedback template elements are zero, i.e., $\mathbf{A} \equiv \mathbf{0}$. In view of (5) each cell of the zero-feedback CNN is described by expression

$$\dot{x}_{ij} = -x_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z. \quad (7)$$

The discrete CNN model is used for image processing. The feedforward CNN description in discrete time domain results from expression (7) after following transformations:

– the approximation of derivative

$$\frac{dx_{ij}(t)}{dt} \approx \frac{x_{ij}(t) - x_{ij}(t - \Delta t)}{\Delta t} = x_{ij}(n) - x_{ij}(n-1)$$

where n is the discrete normalized time. Let us suppose, that $\Delta t = 1$;

– the transition from differential equation (6) to recursive difference equation

$$x_{ij}(n) - x_{ij}(n-1) = -x_{ij}(n-1) + \mathbf{B} \otimes \mathbf{U}_{ij} + z.$$

Eventually, on the bases of (6) and (7) the model of cell C_{ij} in feedforward discrete-time CNN (DTCNN) is described as

$$x_{ij}(n) = \mathbf{B} \otimes \mathbf{U}_{ij} + z,$$

$$y_{ij}(n) = f(x_{ij}(n)). \quad (8)$$

The structure of cell C_{ij} in feedforward DTCNN is depicted in Fig. 3.

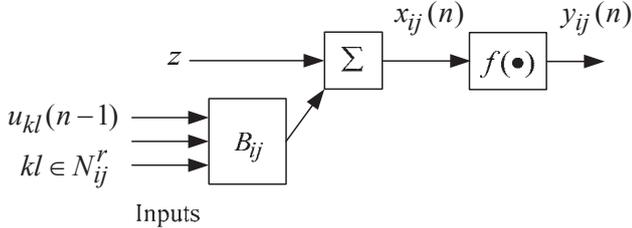


Fig. 3. The structure of cell C_{ij} in feedforward DTCNN

Expressions (8) are transformed into the isotropic model of DTCNN cell if the parameters of model are fixed for entire neural network.

IV. THE SIGNALS OF IMAGES AND THE CRITERION OF FILTRATION ACCURACY ESTIMATION

Combined neural filters are synthesized on the class of bit-map (dot element) half-tone images at the resolution measured by 256 gray levels, i.e., image is the matrix of integers (elements of brightness, pixels) in the interval $[0; 255]$. In the case under consideration, the pixel format is unit8.

The impulse noise represents switched on and switched off pixels (white and black dots in the picture), the emergence of which does not depend on the presence of noise spikes in adjacent dots. The addition of impulse interference to image implies that value q of the signal sample with probability P_a is replaced with value $z = 0$ (black), with probability P_b is replaced with value $z = 255$ (white), and with probability $(1 - (P_a + P_b))$ remains unchanged. Thus, the probability density of the impulse noise is described by the following expression

$$p(q) = \begin{cases} P_a & \text{at } q = 0; \\ P_b & \text{at } q = 255; \\ 0 & \text{in other cases} \end{cases}$$

or

$$p(q) = P_a \delta(q - a) + P_b \delta(q - b),$$

where $\delta(\cdot)$ is the δ -function. Let us assume that $P_a = P_b$. The impulse noise model described above is referred to as “salt and pepper” [4], [9], [10].

In building operator F_ε of nonlinear filter, the “unit8” format of signals $u(n) \in U$ and $y^o(n) \in Y^o$ is transformed into the “double” format (samples of signals are normalized floating-point numbers of double precision in the range $[-1; 1]$).

The synthesis of combined neural filter (Fig. 1) with internal isotropic feedforward DTCNN, including some neurons in hidden layer and referred to as combined DTCNN (CDTCNN), is performed in the MATLAB system environment. This synthesis involves solving the approximation problem (1) in the mean-square norm using the error back propagation algorithm [6]. Activation function f is specified in the form of piecewise linear saturation one (Fig. 2). The distorted image “Tigers” having the size of 220×148 pixels is a learning signal.

The results of CDTCNN filtration are compared with the results of combined neural filtration (Fig. 1) with internal two-layer perceptron network (TLPN), including the hyperbolic tangent activation functions, referred to as combined TLPN (CTLPN) [10], of combined Volterra filtration [5], [6] and of median filtration performed at the 3×3 square aperture [4].

The TLPN model is described by equation

$$y(n) = G \left(c_0 + \sum_{d=1}^I c_d G \left(\sum_{g=0}^m w_{dg} u_g(n) \right) \right),$$

where model input signal is the vector

$$[u_0(n), u_1(n), u_2(n), \dots, u_m(n)],$$

containing $u_0(n) = 1$, as well as elements, corresponding matrix U_{ij} in (4), from the 3×3 window located around cell $C(i, j)$; $m = 9$; G is the hyperbolic tangent activation function; I is the number of neurons.

The structure of two-layer perceptron network is shown in Fig. 4.

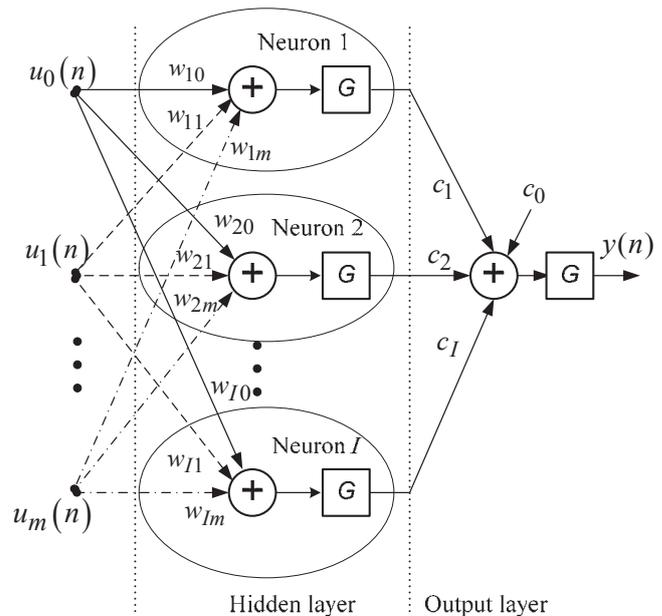


Fig. 4. The structure of two-layer perceptron network

Combined Volterra filter (CVF) is the cascade connection of two units: MF with the 3x3 square aperture and Volterra filter. The CVF structure has the similar form depicted in Fig. 1, but the block NN is replaced with Volterra filter. The CVF model is the truncated Volterra series [5], [6]:

$$y(n) = \sum_{\ell=1}^L \sum_{j_1=0}^{J-1} \sum_{j_2=0}^{J-1} \dots \sum_{j_{\ell}=0}^{J-1} h_{\ell}(j_1, j_2, \dots, j_{\ell}) \prod_{r=1}^{\ell} u(n-j_r), \tag{9}$$

where $h_{\ell}(j_1, j_2, \dots, j_{\ell})$ is the Volterra kernel of order ℓ , $(J-1)$ is the CVF memory length, L is the degree of Volterra model. The set of signals $u(n) \in U$ is formed at the output of smoothing MF and consists of J -length fragments, each of which is built at the aperture moving along the picture with a step of one cycle.

Parameters of the VCF model (9) are defined as a result of solving the approximation problem (1) in the mean-square norm while using learning image with the size of 220×148 pixels. The length of learning sequence $u(n)$ amounts to 32560 samples.

CDTCNN and CTLPN comprise different activation functions. The properties of the piecewise linear saturation function shown in Fig. 2 are considered above.

The properties of the hyperbolic tangent activation function, which is shown in Fig. 5 and contained in CTLPN, are the following:

– gain control for the input signals of different levels. The central function part corresponding to the region of low input signal amplitudes has a large slope and a maximum of the derivative, so the gain is maximum here. Moving from the function central to large absolute values of inputs, the slope of the curve and its derivative is decreased, as well as the gain is reduced;

- the function is continuous and differentiable over the entire range of argument, that is convenient on using this function in the gradient algorithms of learning network when multiple operations of differentiation are required;
- the slow conversion of signal;
- more complicated implementation of the hyperbolic tangent activation function as compared with the piecewise linear function.

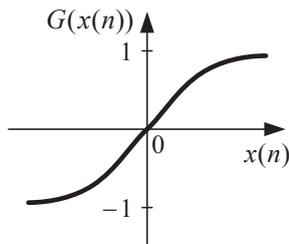


Fig. 5. The hyperbolic tangent activation function

The rate of impulse noise suppression by different filters is estimated with the help of error ε computed after the restoration of a test image having the size of 220×148 pixels by the following formula

$$\varepsilon = \frac{1}{Q} \sum_{n=1}^Q \left(y(n) - y^o(n) \right)^2, \tag{10}$$

where $y(n)$ is the output signal of non-linear filter, $y^o(n)$ is desirable signal, $Q = 32560$. The test images differ from learning one.

V. IMPULSE NOISE CANCELLING BY NON-LINEAR FILTERS

The filtration results (error calculated from (10)) obtained by synthesized devices are summarized in the Table I, the Table II, Fig. 6 and Fig. 7 under various densities p of the impulse noise. “Tigers”, “Building” and “Fence” are the names of learning and two test images, correspondently. All the images have the size of 220×148 pixels.

Neural filters built on the base of CDTCNN and CTLPN comprise two and five neurons in hidden layer under the impulse noise density $p = 0.3$ and $p = 0.5$ correspondingly. The filtration errors are decreased very slowly at neuron numbers higher than mentioned ones.

CVF model is applied in the form of Volterra polynomial (9) of the second degree.

The number of parameters in neural and polynomial models is indicated in the Table III.

TABLE I. FILTRATION ERROR UNDER $P=0.3$

Image	CDTCNN	CTLPN	CVF	MF
Tigers	514	516	534	739
Building	786	804	835	1039
Fence	1142	1138	1153	1434

TABLE II. FILTRATION ERROR UNDER $P=0.5$

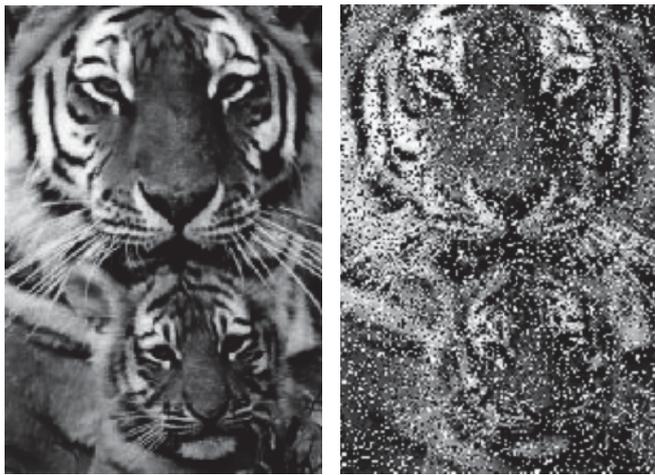
Image	CDTCNN	CTLPN	CVF	MF
Tigers	771	841	1206	2759
Building	1083	1186	1712	3014
Fence	1558	1800	2214	3735

TABLE III. NUMBER OF MODEL PARAMETERS

Model	Impulse noise density	
	0.3	0.5
CDTCNN	23	56
CVF	54	54

The following inferences can be made from the Table I, the Table II and Fig. 6.

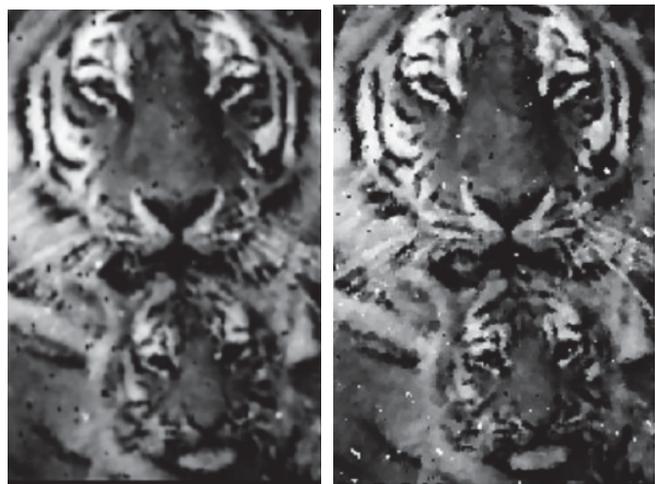
- At the middle noise density ($p = 0.3$), CDTCNN and CTLPN comprise two neurons in hidden layer and 23 parameters. These filters ensure virtually identical accuracy of filtration. Thus, differences in the activation functions of these filters are not revealed in the case of low neuron number.



a b



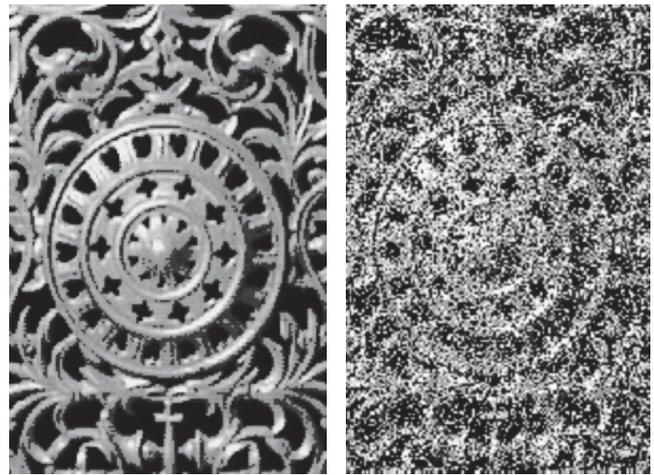
c d



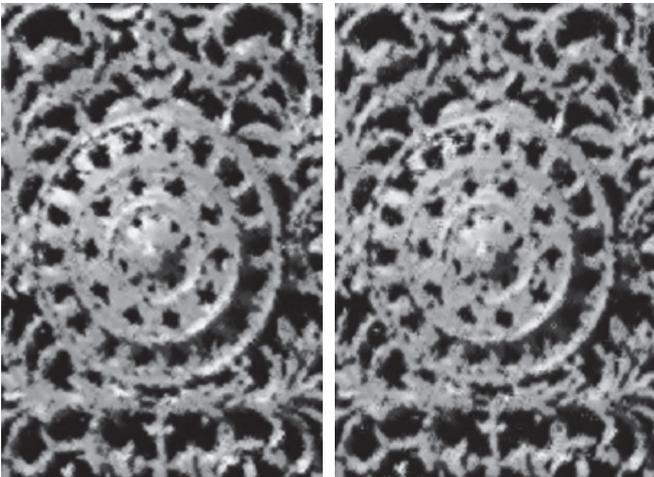
e f

Fig. 6. Learning image and filtration results: a – initial image, b – distorted image, c – CDTCNN result, d – CTLPN result, e – CVF result, f – MF result

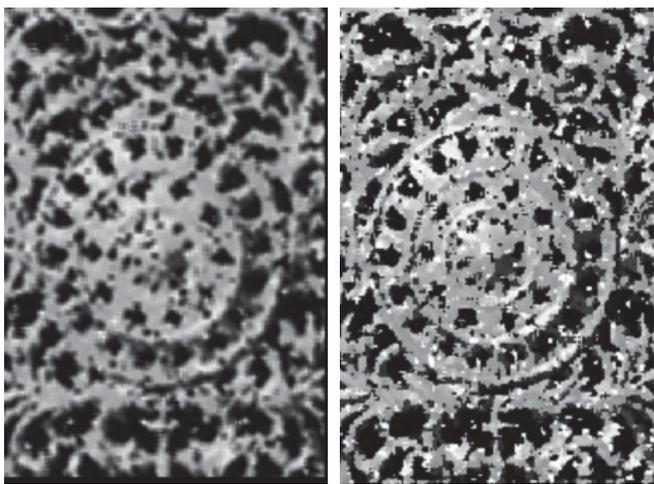
CDTCNN and CTLPN yield higher quality of image restoration (Fig. 6, c, d) than CVF model of the second degree (Fig. 6, e) and MF (Fig. 6, f).



a b



c d



e f

Fig. 7. Test image and filtration results: a – initial image, b – distorted image, c – CDTCNN result, d – CTLPN result, e – CVF result, f – MF result

In practice, CDTCNN is more preferable in comparison with CTLPN since its hardware implementation is simple due to using the piecewise linear saturation functions.

From the Table II, the Table III and Fig. 7, it should be added the following notes.

- At the high noise density ($p = 0.5$), when CDTCNN and CTLPN contain five neurons in hidden layer, CDTCNN with the piecewise linear activation function yields higher filtration precision (Fig. 7, c), than CTLPN with the hyperbolic tangent activation functions (Fig. 7, d), CVF model of the second degree (Fig. 7, e) and MF (Fig. 7, f).

It should be observed that CDTCNN, CTLPN and CVF provide different accuracy at the nearly equal complexity of these filters (56 parameters of CDTCNN and CTLPN, 54 parameters of CVF).

The use of the hyperbolic tangent activation functions in CTLPN negatively affects the image quality (white color turns to gray one, as well as there is a bit ripple i.e. image loses its smoothness (Fig. 7, d)).

Indeed, at an equal probability of the impulse noise (for instance, white and black dots on images) occurrence, the filtration with different gains at low and high amplitudes of signals (in the case of the hyperbolic tangent and the logistic activation function) is not expedient.

VI. CONCLUSION

The problem of non-Gaussian noise filter synthesis is often effectively solved within the framework of the "black box" principle. According to this principle, the mathematical filter model describes the relationship between the sets of input and output signals. The model parameters are determined by solving the approximation problem using the subsets of input and output signals. Considered approach to the synthesis of nonlinear filters is general, because it can be applied at various kinds of non-Gaussian noise sources.

Neural filters such as a feedforward cellular neural network and a two-layer perceptron network surpass Volterra filter and median filter in the accuracy of impulse noise filtration on half-tone images. The above-mentioned neural filters yield equal accuracy at the middle noise density. The feedforward cellular neural network in the standard version (with the piecewise linear saturation functions) carries out more accurate restoration of images as compared with two-layer perceptron network (with the hyperbolic tangent or the logistic activation function) at the high noise density.

It should be emphasized that the hardware implementation of the cellular neural network is simpler in comparison with two-layer perceptron network since the piecewise linear saturation functions used in the cellular network are simpler than the hyperbolic tangent activation functions included in perceptron network.

ACKNOWLEDGMENT

This work was supported by Saint Petersburg Electrotechnical University "LETI" according to the base part of the state scientific work from the Russian Education Ministry.

REFERENCES

- [1] S.V. Vaseghi, *Advanced digital signal processing and noise reduction*. New York: John Wiley & Sons, Ltd, 2008.
- [2] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications*. Kluwer Academic Publishers, 1990.
- [3] S. Haykin, *Digital communication systems*. New York: John Wiley & Sons Inc., 2013.
- [4] R.C. Gonzalez and R.E. Woods, *Digital image processing*. Higher Education, 2008.
- [5] V.J. Mathews and G.L. Sicuranza, *Polynomial signal processing*. New York: John Wiley & Sons, Inc., 2000.
- [6] A. Janczak, *Identification of nonlinear systems using neural networks and polynomial models. A Block-Oriented Approach*. Springer-Verlag Berlin Heidelberg, 2005.
- [7] *Speech, audio, image and biomedical signal processing using neural networks* / Editors: Bhanu Prasad, S.R. Mahadeva Prasanna. Springer-Verlag Berlin Heidelberg, 2008.
- [8] A.A. Lanne and E.B. Solovyeva, "Nonlinear filtration of images with pulse interferences (theoretical principles)", *Radioelectronics and Communications Systems*, vol.43, no. 3, 2000, pp. 546-553.
- [9] A.A. Lanne and E.B. Solovyeva, "Nonlinear filtration of images with pulse interferences (examples of realization)", *Radioelectronics and Communications Systems*, vol.43, no. 4, 2000, pp. 546-553.
- [10] E.B. Solovyeva and S.A. Degtyarev, "Synthesis of neural pulse interference filters for image restoration", *Radioelectronics and Communications Systems*, vol. 51, no. 12, 2008, pp. 661-668.
- [11] L.O. Chua and L. Yang, "Cellular neural networks: Theory". *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, 1988, pp. 1257-1272.
- [12] L.O. Chua and L. Yang, "Cellular neural networks: Applications". *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, 1988, pp. 1273-1290.
- [13] *Chaos, CNN, Memristors and Beyond. A Festschrift for Leon Chua* / Edited by: A. Adamatzky, G. Chen. Singapur: World Scientific Publishing Co. Pte. Ltd., 2013.
- [14] L.O. Chua and T. Roska, *Cellular Neural Networks and Visual Computing: foundations and applications*. Cambridge: Cambridge Univ. Press, 2002.
- [15] M.E. Yalcin, J.A.K. Suykens, and J.P.L. Vandewalle, *Cellular Neural Networks, Multi-Scroll Chaos And Synchronization*. World Scientific Publishing Co. Pte. Ltd., 2005.
- [16] J. Kung, D. Kim, and S. Mukhopadhyay, "On the impact of energy-accuracy tradeoff in a digital cellular neural network for image processing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, 2015, pp. 1070-1081.
- [17] S. Duan, X. Hu, Z. Dong, L. Wang, and P. Mazumder, "Memristor-based cellular nonlinear/neural network: design, analysis, and applications", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, 2015, pp. 1202-1213.
- [18] J. Muller, Jan Muller, R. Braunschweig, and R. Tetzlaff, "A cellular network architecture with polynomial weight functions", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 1, 2016, pp. 353-357.