A Diffusion Based Wave Computing Algorithm for Real Time Edge detection

Mojtaba Karami, Mohammad Rahmati*, Reza Safabakhsh

***Abstract*—** In this paper, a new wave computing algorithm for edge detection in real images is introduced. This algorithm is suitable for real time applications due to the parallel processing capabilities of CNN. The new algorithm is based on the wave computing concept, using diffusion for noise reduction and weak edge elimination and trigger wave to emphasize the strong edges in the image. The proposed algorithm finds edge maps in eight directions and these maps are summed to produce final edge map. The performance of our proposed algorithm is evaluated and compared with different diffusion models using the Berkeley dataset BSDS300 with its benchmark. Experimental results demonstrate superiority of our proposed for real images.

***Keywords*—** Cellular Neural network (CNN), Wave Computing, Diffusion, Real Time Edge Detection.

I. Introduction

CELLULAR Neural Networks (CNNs), introduced by Chua and Yang in 1988 [1][2], consist of a grid of cells with local neighbor interconnections. The CNN Universal Machine (CNN-UM) architecture which is based on CNN structure was proposed by T. Roska and L.O. Chua [3], is suitable for VLSI implementation and is a powerful tool in parallel processing for signal and image processing applications [4][5]. The CNN-based parallel computing relies on analogue signals and connections using templates. This computing approach leads to a processing method called the *Cellular Wave Computing* [6]. Since CNN-UM was introduced, spatio-temporal continuous nonlinear dynamics have been implemented on CNN's and nonlinear partial differential equations (PDEs) have become applicable and new results for different applications based on wave computing on CNN-UM (Cellular Wave Computer [7]) have been achieved. This kind of processing, unlike the Boolean logic, is a spatio–temporal logic defined by spatio–temporal patterns [6]. Using this method, new type of algorithms can be utilized for robot navigation [9], automated detection of a preseizure state [10], target detection [11], object comparison [12], learning of spatio-temporal behavior [13], and auditory scene analysis [14]. In some applications, the method shows the ability to mimic physical phenomena such as the artificial retina model [15][16].

The autowave principle for parallel image processing was proposed by Krinsky [17] and autowave for image processing on two-dimensional CNN was introduced by Perez-Munuzuri [18]. Solving partial differential equations (PDE's) such as reaction diffusion type and systems of ordinary differential equations (ODE's) by CNN have been investigated [19][20]. Diffusion based image processing, as PDE type of autowave computing based on CNN, was discussed by Rekeczky [21]. He investigated PDE-based (constrained linear and nonlinear) diffusion models and a non-PDE based diffusion model, and introduced analogic algorithms for segmentation and edge detection on CNNs. He also investigated the qualitative properties of trigger-wave as a computational tool and its capability in segmentation and shape and structure detection [22].

Previous cellular wave computing studies are called wave computing algorithms [7]. Each step in these algorithms can be seen as a CNN architecture which receives the input and initial state from its previous steps and generates an output for the next steps (CNNs) which is used as the input or initial state. Discretely spaced system of reaction-diffusion employs a mechanism for detecting edges from an image intensity distribution [23][24]. The discrete structure of CNN and its analogic ability is suitable to develop new natural like mammalian vision algorithms [15].

In this paper, a new algorithm for edge detection of gray level images is introduced. The proposed algorithm is simple and straightforward based on diffusion properties. The experimental results demonstrate the superiority of edge detection for real images.

The structure of this paper is as follows. In the next section, the CNN and wave computing are briefly introduced. The proposed algorithm is presented in Section III. Section IV presents experimental results and comparisons with previews works and section V is our conclusion.

II. Cellular Neural Networks and Wave Computing

A standard CNN architecture is a continuous-time network of locally interconnected similar dynamic cells. The cells $C(i,j)$ are arranged in an M×N rectangular array with Cartesian coordinates $(i,j), i = 1,2,\dots,M, j = 1,2,\dots,N$ to form a one-layer CNN (Fig. 1). Neighbors of cell $C(i,j)$

M. Karami is with the Department of Computer Engineering, Amirkabir University of Technology, Tehran 15914, Iran (e-mail: m.karami@ aut.ac.ir).

M. Rahmati is with the Department of Computer Engineering, Amirkabir University of Technology, Tehran 15914, Iran (e-mail: rahmati@ aut.ac.ir).

R. Safabakhsh is with the Department of Computer Engineering, Amirkabir University of Technology, Tehran 15914, Iran (e-mail: safa@ aut.ac.ir).

* Corresponding author.

which are connected to $C(i,j)$ form a set $N_{ij}(r)$ specified by

$$N_{ij}(r) = \{C(k,l) \mid \max(|k-i|,|l-j|) \le r, 1 \le k \le M; 1 \le l \le N\}$$
$$i = 1,2,\dots,M, j = 1,2,\dots,N.$$

where r is a positive integer.

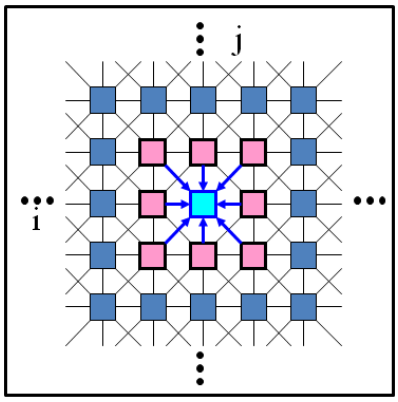

Fig. 1. CNN structure and indicated cell with coordinates (i,j) and its neighbors in sphere with radius r=1

A cell is a dynamical non-linear system with a state equation given by

$$\frac{d}{dt}x_{ij} = -x_{ij} + \sum_{C(k,l)\in N_{ij}(r)} A_{ij,kl}y_{kl}(t) + \sum_{C(k,l)\in N_{ij}(r)} B_{ij,kl}u_{kl}(t) + \sum_{C(k,l)\in N_{ij}(r)} \hat{D}_{ij,kl}\Delta v_{ij,kl}(t) + z_{ij} \quad (1)$$

and the output equation is given by

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

where

$\Delta v_{ij,kl}(t) = v_{kl}(t) - v_{ij}(t)$,

$v_{ij}(t) = u_{ij}(t)$ or $x_{ij}(t)$ or $y_{ij}(t)$

$|x_{ij}(0)| \le 1,\ |u_{ij}(t)| \le 1, |z_{ij}| \le z_{max}$

$,1 \le i \le M\ ,\ 1 \le j \le N$

where u_{ij}, x_{ij}, y_{ij} are the input, state and output voltage of the cell (i,j) in the CNN grid, respectively. $\hat{D}_{ij,kl}$ is the non-linear term that is the difference between a parameter value of each cell with values of its neighbor cells. The parameter can be input, state or output of the cell. If the coefficients, $A_{ij,kl}y_{kl}(t)$, $B_{ij,kl}y_{kl}(t)$ and z_{ij} are space invariant, they make the f*eedback matrix A*, control matrix *B* and bias map *z*. The set $\{A, B, z\}$ is called the cloning template.

The CNN architecture and dynamic behavior of its cells shows the capability of the structure to mimic the brain operation. Based on this, some principles were extracted from neuroscience, genetics and immunology which have been used in CNNs to solve a few difficult problems. These principles include the twin wave principle, push-pull principle, immune response inspired principle, embedded grammar principle, selective spatial modulation principle and fusion of multimodal features [29].

The behavior of dynamic phenomena can be described by continuous time and space partial differential equations. To solve these PDEs using CNN, a PDE is approximated by a grid of cells in which the behavior of each cell is characterized by ordinary differential equations. Each cell works as an ODE solver with continuous state and time and has a sphere of influence of cells. Therefore, the PDE is discretized in space with continuous time. This structure of cells is similar to the nervous organs in creatures.

Wave computing is based on solving the PDE defining a dynamic phenomenon. This point indeed demonstrates the difference between logic computation and wave computing. In digital computers, algorithms are defined on integers, whereas in wave computing, algorithms are defined on solution of a nonlinear wave equation: typically, a reaction diffusion equation. The formal definition of an algorithm in a cellular wave computer is introduced in [29] by Roska.

The CNN-based diffusion models are discussed in by Rekeckey [21] included the PDE-based (constrained linear and non-linear) diffusion approaches (Perona and Malik model, Nordstrom's model) and a Non-PDE approach. In the following the mentioned approaches are reviewed briefly.

A. Constrained linear diffusion

The reaction-diffusion type PDE used by Rekeckey et al. [30] for image processing is in the form of

$$\frac{\partial I(x,y,t)}{\partial t} - div\left(grad\left(I(x,y,t)\right)\right) = \Im_1\left(I_0(x,y,t_0)\right) + \Im_2(I(x,y,t)) \quad (2)$$

where I is the image intensity, I_0 is initial state and $\Im_1(.)$ and $\Im_2(.)$ are nonlinear functions. The parameters x and y are continues coordinate in the two dimension space.

Subclasses of the reaction-diffusion type PDE can be obtained based on different choices of the right hand side of the equation (2). If the right hand side is zero, the *linear diffusion equation* results. By ignoring $\Im_2(.)$, the *constrained linear diffusion equation* is obtained. If $\Im_2(.) = sigm(.)$, the *trigger-wave equation* and if $\Im_2(.) = sigm(.)$ and $\Im_1(.) \neq 0$, the *constrained trigger-wave equations* will result.

To map the above PDE on CNN it should to discretize the x and y along the two dimensions in space. The discretization of the mentioned parameters, the following reaction-diffusion type nonlinear ODE is acquired [29][30][31]. Two parameters i and j is used for the discretization of x and y,

$$\frac{dI_{ij}(t)}{dx} = g\left(I_{ij}(t)\right) - I_{ij}(t) + \frac{c_1}{4}\left(y_{i-1j}(t) + y_{i+1j}(t) + y_{ij-1}(t) + y_{ij+1}(t)\right) + z_{ij} \quad (3)$$

and

$y_{ij}(t) = f(I_{ij}(t))$, $g(.) = c_0 f(.)$,

$z_{ij} = z_0 + \sum_{C(k,l)\in N_{ij}(r)} b_{kl}\, I_{kl}(t_0)$

where z_{ij} and g(.) take the places of $\Im_1(.)$ and $\Im_2(.)$ in Eq. (2), respectively. The N_{ij} defines the nearest neighbors for cell

(i,j). By using $z_{ij}=0$ and $f(I)=I$, the linear diffusion equation, $z_{ij}\neq 0$ and $f(I)=I$, the constrained linear diffusion equation, $z_{ij}=0$ and $f(I)=sigm(I)$, the nonlinear trigger-wave equation, and $z_{ij}\neq 0$ and $f(I)=sigm(I)$, the constrained nonlinear trigger-wave equation will result.

The equivalent CNN templates for the diffusion filter and trigger wave have the following form

$$A=\begin{bmatrix}0 & a_1 & 0\\ a_1 & a_0 & a_1\\ 0 & a_1 & 0\end{bmatrix},\ B=\begin{bmatrix}b_2 & b_1 & b_2\\ b_1 & b_0 & b_1\\ b_2 & b_1 & b_2\end{bmatrix},\ z=z_0 \tag{4}$$

By setting the parameters $a_0=0$ and $a_1>0$, the cloning template $\{A,B,z\}$ is called the diffusion. By setting the parameters $a_0>a_1>0$, the cloning template $\{A,B,z\}$ is called the **trigger wave**. Diffusion and trigger wave have been used in some applications such as texture segmentation and detection, change detection in video flow, object comparison, contrast enhancement, noise suppression and shape enhancement [32].

By setting the parameters $a_0=0$ and $a_1=0.25$, the diffusion cloning template acquired, we used it in experiments and for simplicity called it **Diffus :**

$$A=\begin{bmatrix}0 & 0.25 & 0\\ 0.25 & 0 & 0.25\\ 0 & 0.25 & 0\end{bmatrix},\quad B=0,\quad z=0 \tag{5}$$

If the $\mathfrak{J}_1(.)$ is defined as a simple low-pass filter and by setting the parameters $a_0=0$ and $a_1=0.25$, a constrained linear diffusion will obtained, we called it **CDiffus** in this paper.

$$A=P\begin{bmatrix}0 & 0.25 & 0\\ 0.25 & 0 & 0.25\\ 0 & 0.25 & 0\end{bmatrix},$$
$$B=(1-P)\begin{bmatrix}0 & 0.25 & 0\\ 0.25 & 0 & 0.25\\ 0 & 0.25 & 0\end{bmatrix},\quad z_0=0 \tag{6}$$

P is the diffusion scale factor. If $P=1$, **CDiffus** reduces to **Diffus** and if $P=0$, the convolution is obtained.

B. Anisotropic diffusion

Perona and Malik [33] proposed the anisotropic diffusion formulation as :

$$\frac{\partial}{\partial t}I(x,y,t)=div\left(c(x,y,t)grad\big(I(x,y,t)\big)\right) \tag{7}$$

and

$$I(x,y,t_0)=I_0(x,y),$$
$$c(x,y,t)=\left(1+\left(\left|grad\big(I(x,y,t)\big)\right|/K\right)^{1+\alpha}\right)^{-1}$$

where $I(x,y,t)$ is the image intensity, $I_0(x)$ the initial state and parameters x and y are continues coordinate in a two dimension space.

The spatial discretization of the above formulation, the following CNN cell model in the linear region of the piecewise linear function $f(.)$ obtained [33] :

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{I_{ij}(t)}=-\mathrm{I_{ij}(t)}+\mathrm{a_0y_{ij}(t)}+\sum_{\mathrm{C(k,l)\in N_{ij}(r)}}\widehat{\mathrm{D}}_{\mathrm{ij,kl}}\Delta\mathrm{v_{ii}}+\mathrm{z_{ij}} \tag{8}$$
$$y_{ij}(t)=f\left(I_{ij}(t)\right),\qquad \Delta\mathrm{v_{ii}}=I_{kl}(t)-I_{ij}(t)$$
with
$$a_0=1,\quad \widehat{\mathrm{D}}=\begin{bmatrix}0 & \Phi & 0\\ \Phi & 0 & \Phi\\ 0 & \Phi & 0\end{bmatrix},$$
where $\Phi=g\Delta\mathrm{v_{ii}}$, $\mathrm{g}=\begin{cases}1-|\Delta\mathrm{v_{ii}}|/2\mathrm{K} & \text{if } |\Delta\mathrm{v_{ii}}|<2K\\ 0 & \text{otherwise}\end{cases}$
and $\mathrm{z_{ij}}=0$.

We called this formulation and the corresponding template as **PM** model**.**

C. Constrained anisotropic diffusion

This approach is a generalization of the model proposed by Nordstrom [21]. The formulation of the mentioned approaches is as :

$$\frac{\partial}{\partial t}I(x,y,t)-div\left(c(x,y,t)grad\big(I(x,y,t)\big)\right)=\beta\big(I(x,y,t_0)\big)-I(x,y,t) \tag{9}$$

and $I(x,y,t_0)=I_0(x,y)$.

By assuming $\beta\big(I(x,t_0)\big)$ as the convolution of a Gaussian with the input image, denoted as $\beta\big(I(x,y,t_0)\big)=G_\sigma * I(x,y,t_0)$, the CNN cell model corresponding to the formulation is same as Equation (8) with templates give by :

$$a_0=1,\ \widehat{\mathrm{D}}=\begin{bmatrix}0 & \Phi & 0\\ \Phi & 0 & \Phi\\ 0 & \Phi & 0\end{bmatrix},\ \Phi=g\Delta\mathrm{v_{ii}},$$
$$\mathrm{g}=\begin{cases}1-|\Delta\mathrm{v_{ii}}|/2\mathrm{K} & \text{if } |\Delta\mathrm{v_{ii}}|<2K\\ 0 & \text{otherwise}\end{cases},$$
$$\mathrm{B}=\begin{bmatrix}0.05 & 0.15 & 0.05\\ 0.15 & 0.20 & 0.15\\ 0.05 & 0.15 & 0.05\end{bmatrix},\ \mathrm{z_{ij}}=0 \tag{10}$$

In this paper we called this formulation as Nordstrom's model, briefly **N** model**.**

III. PREVIOUS WORKS

In cellular wave computing algorithms, the CNN is used as a computing device which incorporated cellular wave computing algorithms. Each step in these algorithms could be seen as one module (CNN) which receives input and initial state from the previous steps and generates output for the next steps (CNNs).

The first Edge detection algorithm based on CNN was presented by Rekeczky [21]. He investigated PDE-based constrained linear and nonlinear diffusion models and a non-PDE based diffusion model and introduced analogic algorithms for edge detection on CNNs. The edge detection method proposed by Rekeczky consists of three computational blocks: (i) linear or non-linear diffusion based pre-filtering

(Diffusion Filter in Fig. 2), (ii) local threshold estimation demonstrated (Threshold Estimation in Fig. 2), (iii) locally adaptive segmentation and binary post-processing demonstrated as Edge Detection in Fig. 2.

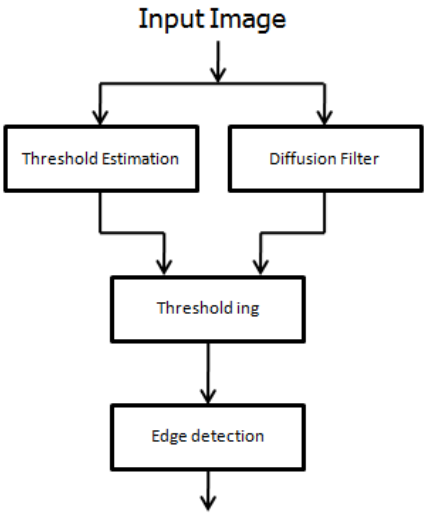

Fig. 2. Flowchart of the algorithm proposed by Rekeczky [21]

In the algorithm presented in Fig. 2 diffusion filter designed for noise reduction and edge enhancement. Threshold estimation is calculated using equation (11) as a space-variant parameter that is a linear combination of the first-order (mean) and second-order (variance) local statistics in the fixed neighborhood radius r scaled by the parameters (α, β). In implementation, the mean and variance are approximated using **CDiffus** (with $P = 0.8$) and Laplace templates respectively and setting $(\alpha = \beta = 0.2)$.

$$\theta = \alpha\varepsilon_r(I) + \beta var_r(I) \tag{11}$$

where I is the input image, ε the mean, var the variance and θ is the threshold map.

The thresholding step uses the space-variant threshold estimation and perform threshold on the output of the diffusion filter. Edge detection stages, are performed using implementation for edge detection in gray level images and is presented in MatCNN toolbox [34]. The edge detection template used in this method is :

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \mathrm{B} = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix},\ z = -1.5$$

Although the edge detection algorithms like algorithm proposed by Canny [35], successfully detects edges, however there are other methods to mimic the human vision systems for edge detection. One of these approaches is based on reaction-diffusion system. A reaction-diffusion system refers to a system consisting of diffusion processes coupled with reaction processes. Discretely spaced system of reaction-diffusion has a mechanism detecting edges from an image intensity distribution [23][24].

Reaction-Diffusion based edge detection algorithms were presented in [23][24][36][37]. These works used a discretely spaced system of the reaction-diffusion equations for edge detection in gray level images. Nomura et al. presented a reaction-diffusion algorithm for edge detection for gray level images with a variable threshold level [36]. The shortcoming of their proposed algorithm is how to eliminate false pulses. This shortcoming was resolved by another edge detection algorithm [37]. Based on the works proposed in [36], P. H. Long et al proposed two architectures for edge detection of real images proposed based on FitzHugh-Nagumo (FHN) reaction-diffusion equation [38]. The CNN structure for edge detection of real images consists of three layers shown in Fig. 3 and the FHN equation used for edge detection is in the discrete form are given as :

$$\begin{aligned} \frac{du_{ij}}{dt} &= D_u\nabla^2 u_{ij} + \frac{1}{\varepsilon}\left[u_{ij}(u_{ij} - a)(1 - u_{ij}) - v_{ij}\right] \\ \frac{dv_{ij}}{dt} &= D_v\nabla^2 v_{ij} + u_{ij} - bv_{ij} \\ \frac{da_{ij}}{dt} &= D_a\nabla^2 a_{ij} \end{aligned} \tag{12}$$

D_u and D_u are diffusive coefficients of variable u and v, respectively. In (12) a and b are constant and ε is a small positive constant $0 < \varepsilon \leq 1$.

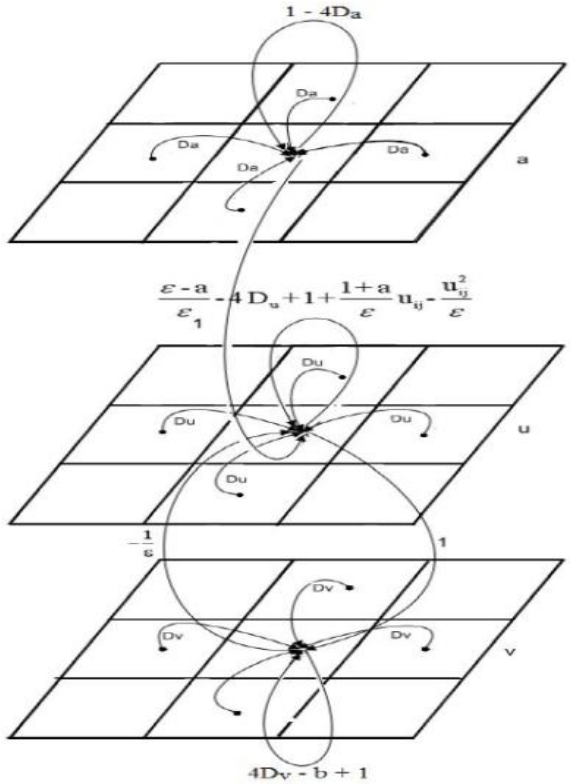

Fig. 3. Tree layer CNN for CNN-FHN model [38]

IV. WAVE COMPUTING ALGORITHM FOR EDGE DETECTION

In this section, we propose a new wave computing algorithm for edge detection which has been applied to gray level images. The proposed algorithm is a simple real time algorithm that its capability is demonstrated using standard

real images data set and comparing with previous wave computing algorithms.

The proposed algorithm is based on diffusion using CNN linear and non-linear templates. The proposed algorithm uses the diffusion as preprocessing stage for noise reduction and edge enhancement. Then using shift in different directions, eight similar images are produced, except for slight differences in some directions. These eight images are called *Shifted Diffusion*. The features in the outputs of the preprocessing stage of our algorithm are used for edge detection. In the next stage of the algorithm, the eight difference images are computed using the differences between the eight shifted images and diffusion image. The difference between a *shifted diffusion* image and diffusion image is used as differentiation at corresponding direction and can be used as edge map of the image in that direction. Using diffusion in the first stage, noise and most of weak edges to be reduced in the output *Shifted Diffusion* images. In the next stage we used the trigger to emphasize the edges in the *Shifted Diffusion* images. As illustrated in Fig. 4, the eight difference outputs from the differentiation stage were used as the inputs to eight standard CNNs using the trigger template to generate trigger wave and extract the edge in eight differences images in eight directions. The triggered output images are fed to eight CNNs to run erosion to reduce noise and thinning the edges in eight directions. The outputs of the erosion stage sum up to produce the final output as the detected edges of the input image.

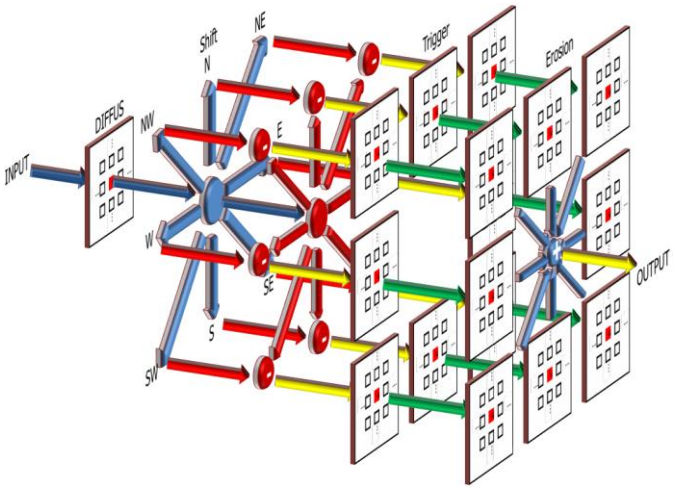

Fig. 4. Wave computing algorithm for boundary edge detection

The trigger template is used in the proposed approach is as the following with $a = 0.25$ and $b = 1$ [34],

$$A = \begin{bmatrix} 0 & a & 0 \\ a & b & a \\ 0 & a & 0 \end{bmatrix},\ \mathrm{B} = \ 0, z = 0\ .$$

The erosion template is used in the proposed approach is as[34] :

$$A = \ 0, B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix},\ z = -4\ .$$

Fig. 5 illustrates an example yield by our algorithm at the output of each layer. In Fig. 5b, the eight outputs of the trigger stage is shown. In these images, the edges in each direction are detected at the corresponding output. The result of running erosion is illustrated in Fig. 5c. The erosion reduces the noise and thinning the triggered edge in the previous stage. The output of the algorithm is the aggregation of the outputs acquired from the erosion stage by adding the edges in the eight directions. The output of the algorithm has less noise than other edge detection wave computing algorithms but the edge map yields thicker edges.

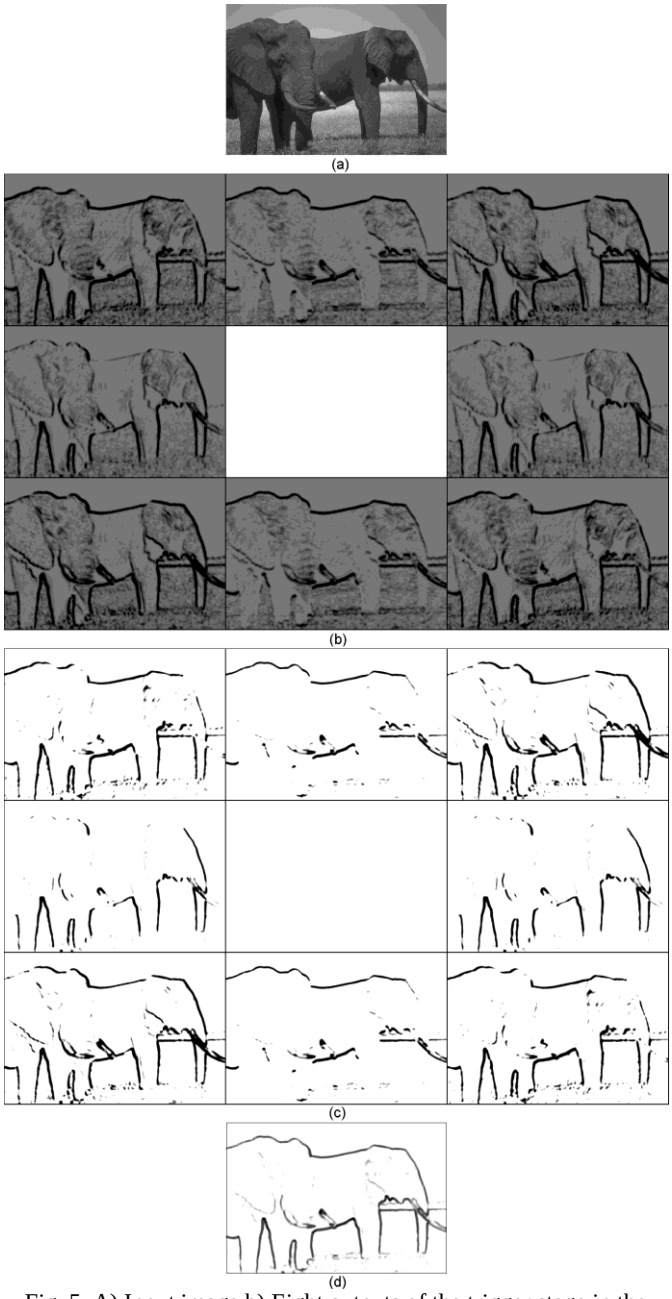

Fig. 5. A) Input image b) Eight outputs of the trigger stage in the proposed algorithm c) Eight outputs of the erosion stage in our proposed algorithm d) Output image edge resulted by adding the images in (c).

V. Experimental Results

The performance of our proposed edge detection is evaluated by comparing it with the algorithm presented by Rekeczky [21], and CNN-FHN model. The dataset used in these experiments, is the Berkeley segmentation dataset (BSDS300)[25] using the benchmark presented by Martin et al. [27][28]. The Berkeley segmentation dataset (BSDS300) is a well-known dataset for evaluating new edge detection and segmentation algorithms [25][26]. This dataset consists of 200 training and 100 testing images each with multiple ground-truth boundaries marked by different peoples. This dataset has been used for evaluating new boundary detection and segmentation algorithms [26].

To evaluate our algorithm on BSDS300 and make the comparisons, the benchmark presented by Martin et al. [27][28] is used. This benchmark produces a score for boundary detection algorithm. The benchmark operates on a non-thresholded boundary image using levels, and finds the optimal threshold. At each threshold level, precision and recall are computed, and a precision-recall curve is produced.

Precision is the probability that a boundary pixel is a true boundary pixel and recall is the probability that a true boundary pixel is detected. In other words, precision is a measure of how much noise is present in the output of the detector and recall is a measure of how much of the ground truth is detected. To compare two algorithms, the F-measure is defined on all points of the precision-recall curve and the maximum F-measure value is reported for each algorithm [26]. The F-measure is defined as

$$F = \frac{2*Precision*Recall}{Precision+Recall}$$

In experiments, we run the algorithm presented by Rekeczky and our algorithm in this paper using different diffusion methods and different parameter values. Also we run CNN-FHN model using different parameter values to find best result. The templates corresponding to linear diffusion are variants of **CDiffus** and can be tuned through two parameters P and transient length of the CNN. By setting P=1 ,the simplest diffusion operation obtained. **CDiffus** with P=0.5 is a typical constrained diffusion ('robust diffusion' [39]) template. By decreasing the weights P to zero the convolution template will be obtained. Non-linear diffusion models **N** and **PM** derived from the PDE-related non-linear models are tuned by two parameters K and the transient length of the CNN. In all experiments the transient length of the CNN is fixed at 5.

We run the above methods using 100 test images in BSDS300 and evaluate the result by the Berkeley benchmark algorithm. Iso-F curves show the precision-recall relation for specific values of the F-measure. Fig. 6 shows the precision-recall curves for 0.1 to 0.9 values of F-measure. Different values of recall-precision on a specific curve have same F-measure. As seen in the Fig. 6, increasing recall causes precision to decrease and vice versa. Edge detection approaches have been ranked according to their maximum F-measure with respect to human ground-truth boundaries. The F-measure according to human ground-truth is illustrated in Fig. 6 as a isolated spot with F=0.79. Using Iso-F curves, new algorithms can be evaluated and compared together and to the human ground-truth.

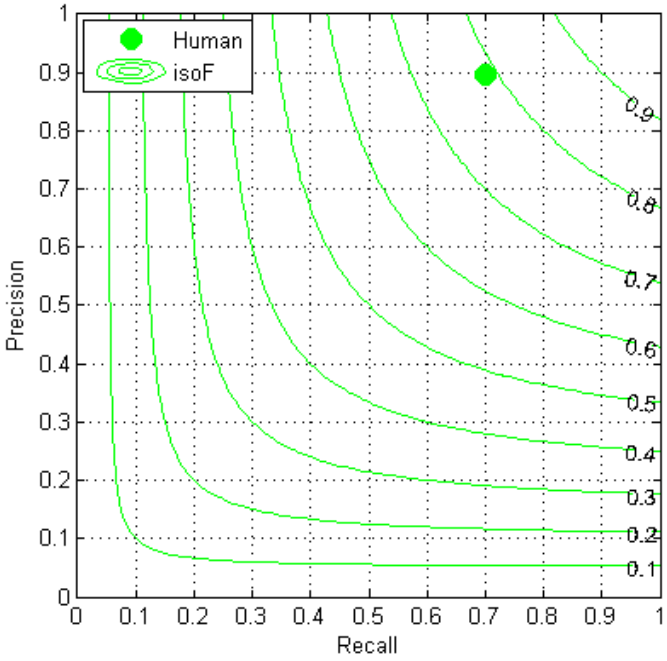

Fig. 6. Iso-F curves according to F-measure between 0.1 to 0.9 and human ground-truth F-measure using 100 testing images in Berkeley segmentation dataset (BSDS300 boundary edge detection, each with multiple ground-truth boundaries marked by different humans

Fig. 7 illustrates the precision-recall curve and the maximum F-measure for proposed algorithm using **CDiffus** setting P from 0.50 to 0.95. As seen in Fig. 7, the proposed edge detection algorithm has F-measures between 0.33 and 0.53 depending on different setting of P. The best result are obtained setting P=0.75 with F=0.53 and (recall ,precision) = (0.57, 50). In Fig. 8, the experiments result using Nordstrom's model (**N**) is illustrated. As seen in this figure, decreasing K beyond 0.2 has no significant effect in the precision and recall parameters. The best F-measure evaluated by benchmark is 0.44 with (recall ,precision) = (0.54, 37) at threshold t=0.98. Experiments using **PM** diffusion model is shown in Fig. 9. By setting K from 0.1 to 1.0 and testing the result, the same F-measure 0.50 obtained for K from 0.4 to 0.9. But by increasing the K, recall measure decreases and precision is growing. The K parameter can be set for desire precision-recall. We chose the K=0.7 that has the median precision-recall as the best result of this experiment for future comparisons.

Fig. 10 illustrates the comparison between best results of proposed method using different diffusion models. The proposed approach using **CDiffus** diffusion model might have an F-measure between 0.05 to 0.6, depending on the desired precision and recall. The best F-measure in this case is 0.53 with (recall ,precision) = (0.57, 50) at threshold t=0.24. The proposed approach using Nordstrom's model (**N**) might have an F-measure between 0.3 to 0.5, depending on the desired precision and recall and the best F-measure is 0.44 with (recall ,precision) = (0.58, 36) at threshold t=0.75. Using Perona and

Malik's model for diffusion in proposed algorithm has the interval 0.1 to 0.5 for F-measure and the best F-measure is 0.50 with (recall ,precision) = (0.44, 58) at threshold t=0.24. As seen in Fig. 10 by using **PM** model in algorithm, it might have higher precision than using other two diffusion models. By using **CDiffus** model in algorithm, it might acquire higher recall than using other two diffusion models. But the using **CDiffus** model has better F-measure than using **PM** model.

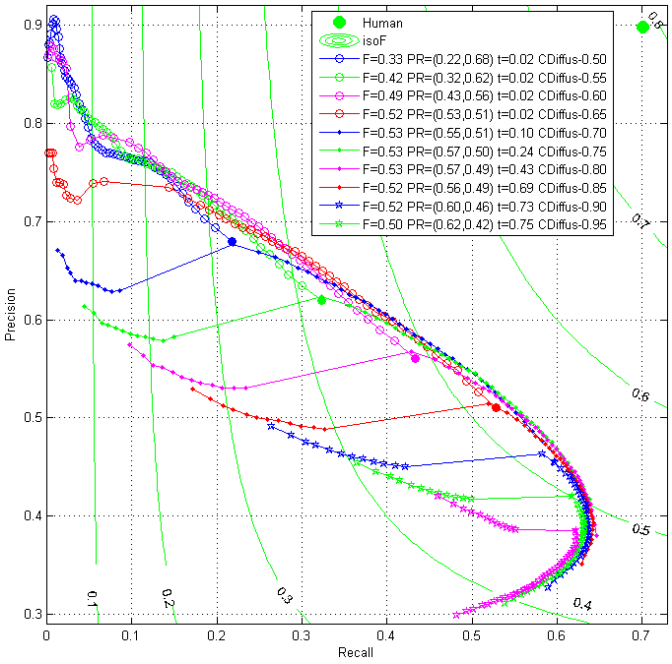

Fig. 7. Proposed algorithm using CDiffus setting P=0.50-0.95

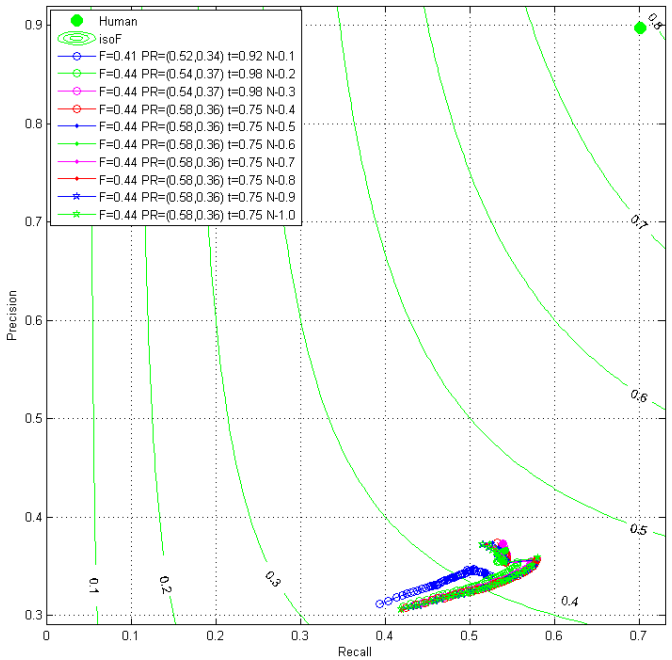

Fig. 8. Proposed algorithm using Nordstrom's diffusion model (N) settingK=0.1-1.0

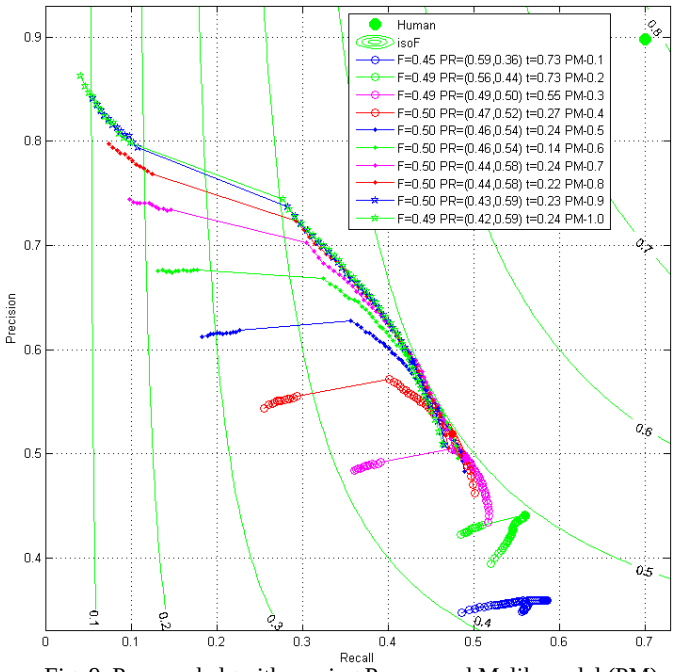

Fig. 9. Proposed algorithm using Perona and Malik model (PM) setting K=0.1-1.0

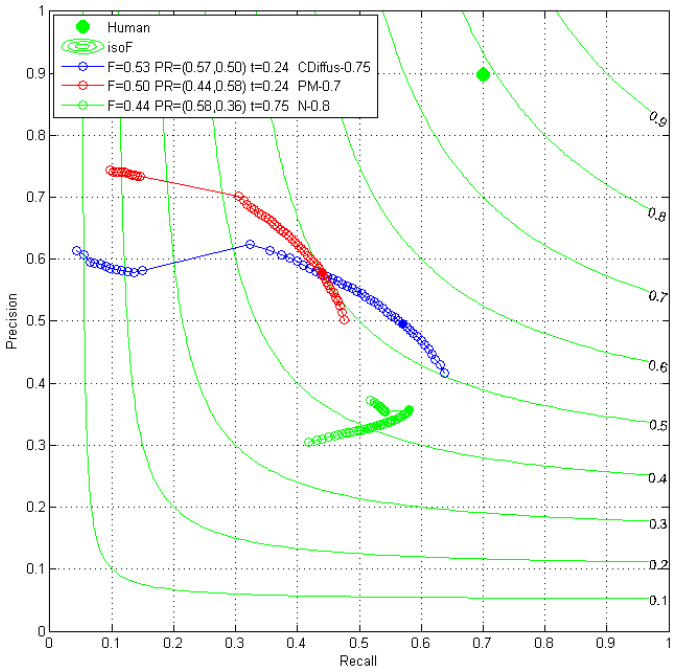

Fig. 10. Comparison of best results of the proposed algorithm using different diffusion models

Similar experiments have been performed by Rekeczky algorithm. In these experiments the F-measure curves are reduced to one spot due to the binary output of the algorithm. In experiments using Rekeczky's algorithm, the threshold estimation equation (11) is used, the mean and variance are approximated using **CDiffus** (with $P = 0.8$) and Laplace templates respectively and setting $(\alpha = \beta = 0.2)$ similar to Rekeczky reported in experiments [21].

Fig. 11 illustrates the precision-recall and the maximum F-measure for Rekeczky's algorithm using **CDiffus** setting P from 0.1 to 1.0. As seen in Fig. 11, the Rekeczky's edge detection algorithm has F-measures between 0.39 and 0.40 depending on different setting of P. We chose the best result setting P=0.1 with F=0.40 and (recall ,precision) = (0.46, 35).

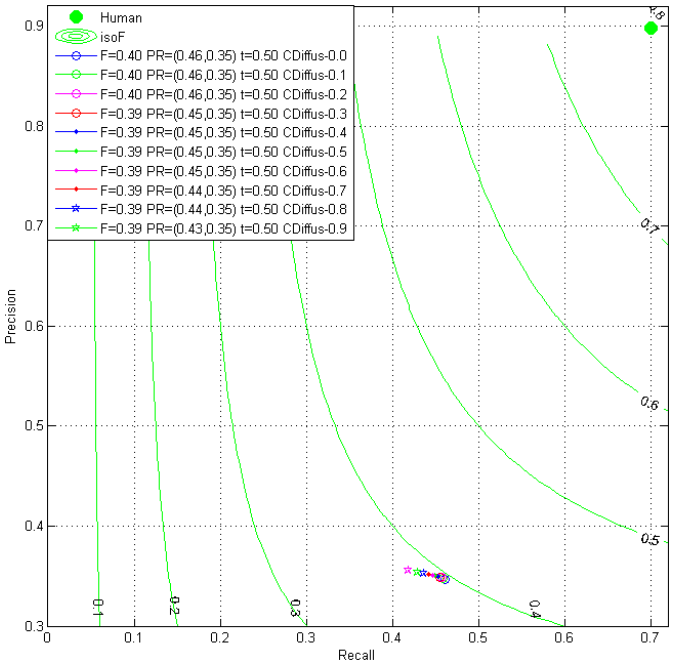

Fig.11. Rekeczky's algorithm using CDiffus setting P=0.0-0.9

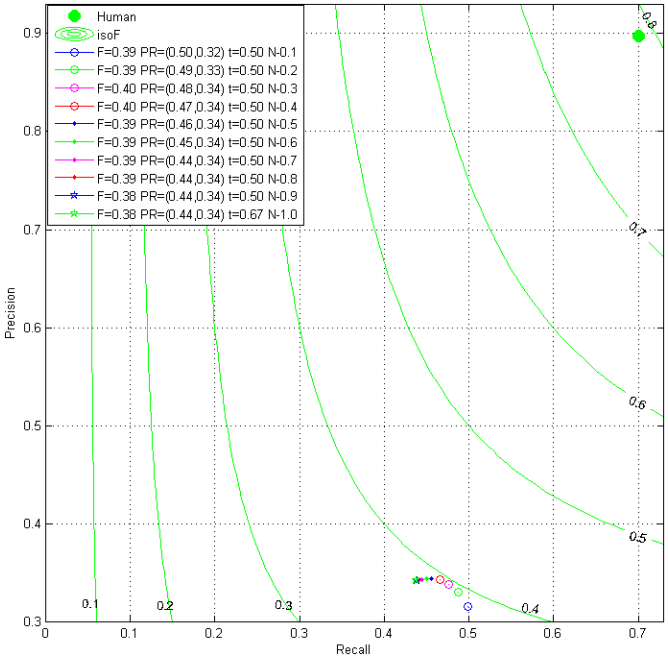

Fig.12. Rekeczky's algorithm using Nordstrom's diffusion model (N) setting K=0.1-1.0

In Fig. 12, the experiments results using Nordstrom's model (**N**) is illustrated. The best F-measure is 0.40 by setting K=0.3 with (recall ,precision) = (0.48, 34) at threshold t=0.50. Experiments using **PM** diffusion model is showed in Fig. 13. By setting K from 0.1 to 1.0, the best F-measure=0.40 is obtained for K=0.2 with (recall ,precision) = (0.46, 35). Fig. 14 illustrates our comparison between the best results of Rekeczky's method using different diffusion models. The Rekeczky's approach using **CDiffus,** Nordstrom's diffusion model (**N**) and Perona and Malik's diffusion model (**PM**) has the same results and there is no significant preference for each one.

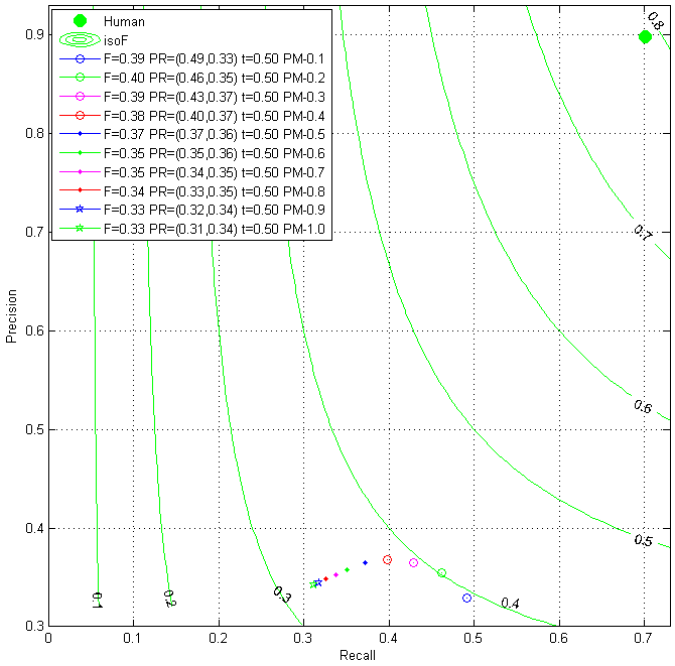

Fig.13. Rekeczky's algorithm using Perona and Malik model (PM) setting K=0.1-1.0

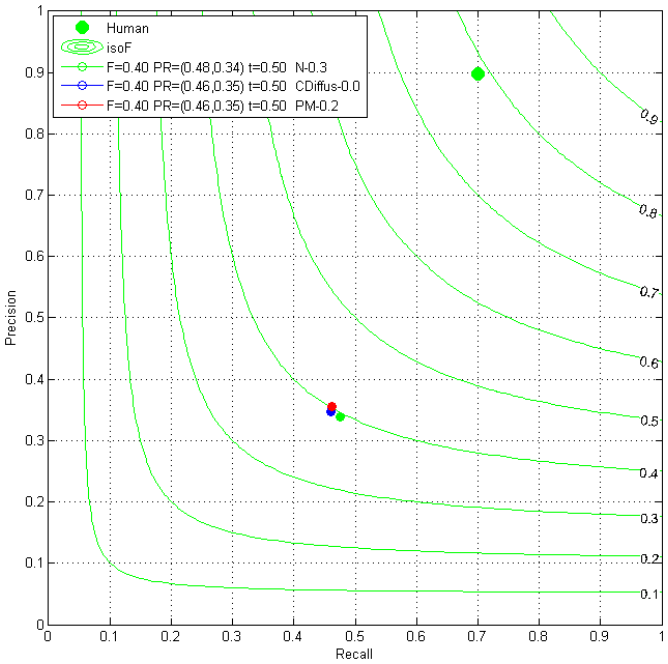

Fig.14. Comparison of best results of the Rekeczky's algorithm using different diffusion models

To evaluate the FHN reaction-diffusion model in real time edge detection we use the parameter values reported in [38] for starting point in our experiments. These parameters are :

$D_u = 1$, $D_v = 5$, $D_a = 50$, $\varepsilon = 0.001$, $b = 50$, $\Delta t = 0.001$, $t = 0.03$, $a_0(i,j) = U_0(i,j)$

Based on the stability conditions of the FHN reaction-diffusion model reported in the references [36][38], diffusive coefficient must satisfy the condition $D_u \ll D_v$. Therefore, diffusive coefficients (D_u and D_v) and timing parameters (Δt and t) for running CNN are selected according to the reported values in [38]. Therefore, there are three free parameters (D_a, ε and b) that the FHN reaction-diffusion model should chose.

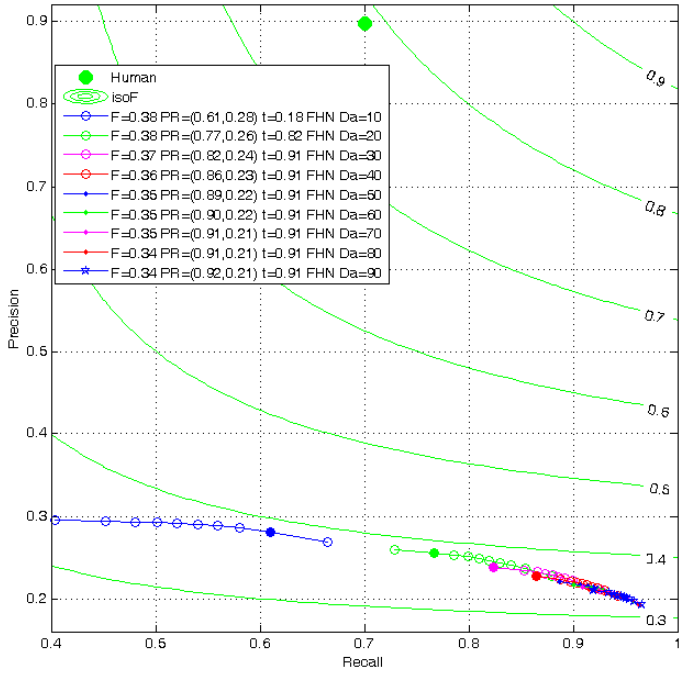

Fig.15. CNN-FHN model using $D_a = 10 - 90$

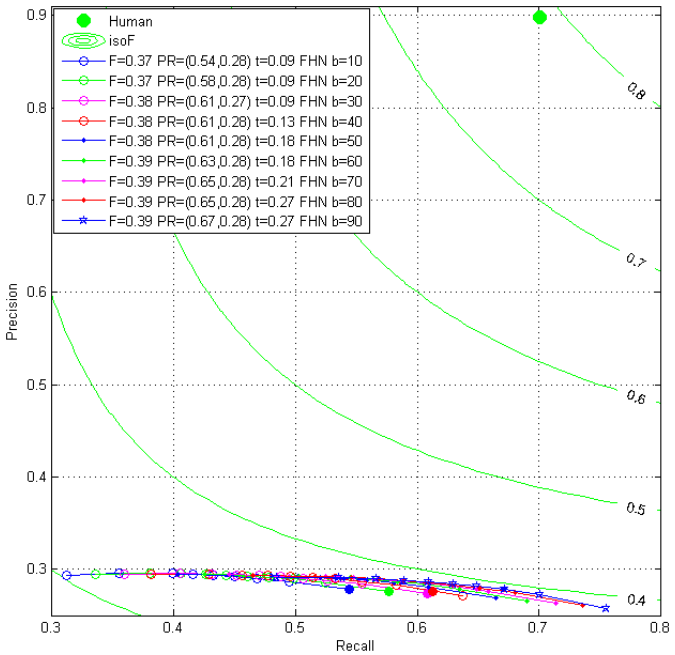

Fig.16. CNN-FHN model using b=10-90

Using these parameters, experiments run at tree stages. At the first stage we chose D_a as the variable (10 to 90) and fixed the other parameters. As seen in Fig. 15, the CNN-FHN model has F-measures between 0.34 and 0.38 depending on different setting of D_a. The best result are obtained setting $D_a = 20$ with F=0.38 and (recall ,precision) = (0.77, 26). Then we fixed $D_a = 20$ and chose b as variable parameter in the domain 10 to 90. In Fig. 16, the experiments result using b as variable is illustrated. As seen in this figure, the best F-measure evaluated by benchmark is 0.39 with (recall ,precision) = (0.67, 28) using $b = 90$. In the next experiment using fixed parameters $D_a = 20$, $b = 90$ and setting ε from 0.001 to 0.009 and testing the result, the best F-measure 0.38 obtained for $\varepsilon = 0.002$ (Fig. 17).

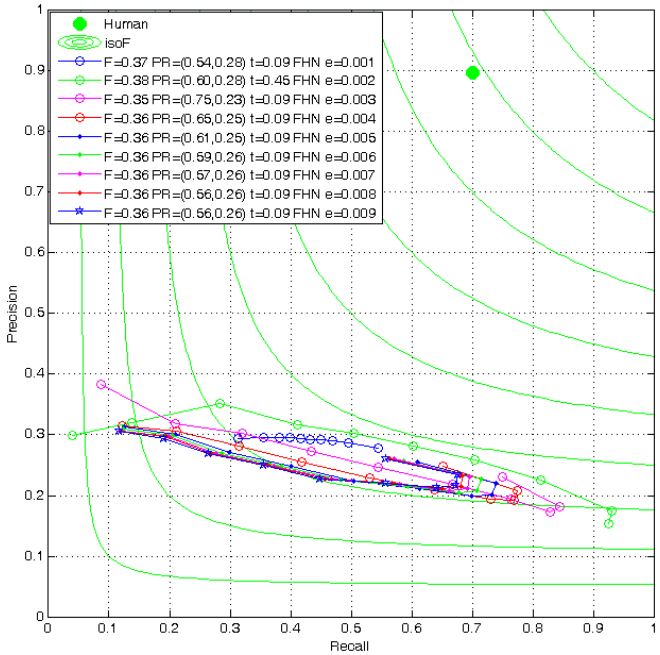

Fig.17. CNN-FHN model using $\varepsilon = 0.001 - 0.009$

In Fig. 18 comparison between the best result of our proposed method, Rekeczky algorithm and FHN edge detector is made. Due to binary output of the Rekeczky's algorithm, this method has only one spot demonstrates the recall-precision and consequently an F-measure. The best experiment of the proposed method, as illustrated in Fig. 10, is by using the **CDiffus** setting K=0.75 with F-measure=0.53 and (recall, precision) = (0.57, 50) at threshold t=0.24. The Rekeczky's algorithm has F-measure= 0.40 with (recall, precision)=(0.48,34) that is lower than our result. The CNN-FHN model has F-measure=0.38 with (recall ,precision) = (0.60, 28). The high recall is acquired by FHN edge detector method show the performance of this detector in finding almost all edges in every image in the dataset. But the low precision show the high rate noise in the result of the algorithm that can be seen in the Fig. 19.

In Fig. 19, five images selected from the DBSD300 with different details and objects, are used to demonstrate the performance between CNN-FHN model, proposed algorithm and Rekeczky's algorithm. The human ground-truth edge detected for selected images is also presented in row b. The results of CNN-FHN model is shown in row c. The best result

of proposed algorithm using three diffusion models **CDiffus**, **PM** and **N** are presented in rows d, e and f respectively. The best result of Rekeczky's algorithm using three diffusion models **CDiffus**, **PM** and **N** are presented in rows g, h and i respectively. Clearly, the proposed method illustrates superior performance than the Rekeczky algorithm. As seen in this figure, the CNN-FHN model find more edges in each image, but it has significant noise. The proposed method show better performance in finding object boundary in the images when using **CDiffus** and **PM** as diffusion models and the results have significant noise using **N** diffusion model based on comparison presented in Fig. 10. As seen in Fig. 10, the recall for algorithm using **N** model is better than using **PM** model, but its precision is low. The proposed algorithm using **N** model as diffusion model works close to CNN-FHN model. The proposed method, using **CDiffus** and **PM** as diffusion models, show good result in noisy images like right hand side image of snake in desert sand. Comparison between proposed method, using **CDiffus** and **PM** as diffusion models, illustrates that using **PM** as diffusion model results images with lower noise but with lower true detected edge. In the other word using **PM** model presents lower recall and higher precision compare to proposed algorithm using **CDiffus** diffusion model (see Fig. 10). Rekeczky's algorithm using three diffusion models **CDiffus**, **PM** and **N** show the same result as we expected based on values in the Fig. 14. In Fig. 14, the same F-measure and closely precision and recall values for Rekeczky's algorithm using these there diffusion models is easily seen.

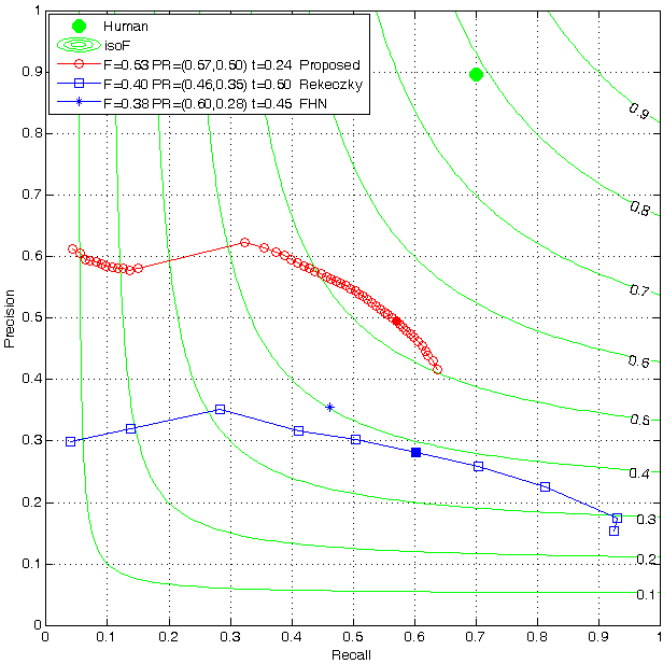

Fig. 18. Comparison of Proposed , Rekeczky and CNN-FHN models

VI. Conclusions

In this paper, we introduced a new wave computing algorithm based on diffusion using CNN linear and non-linear templates. Our proposed algorithm is based on diffusion which is used to reduce the noise. Output of diffusion is used to produce eight shifted images. Difference of shifted images and diffusion image show their usefulness characteristics for edge detection in eight directions. To extract the edges in eight directions, the trigger wave template is run on eight shifted images. Then erosion template is run to reduce the noise and thinning the detected edges. The output of the algorithm is the sum up of the eight erosion images. To show the performance of our algorithm, three diffusion models are reviewed and used in experiments. Several experiments were made running proposed algorithm and edge detection algorithm was proposed by Rekeczky using three diffusion models. Experiments using the BSDS300 and Berkeley evaluating benchmark showed the good performance of the proposed algorithm compare to CNN-FHN model and Rekeczky's algorithm in real edge detection. Our future work will concentrate on developing the proposed algorithm using genetic programming to develop new algorithms using shifted diffusion images.

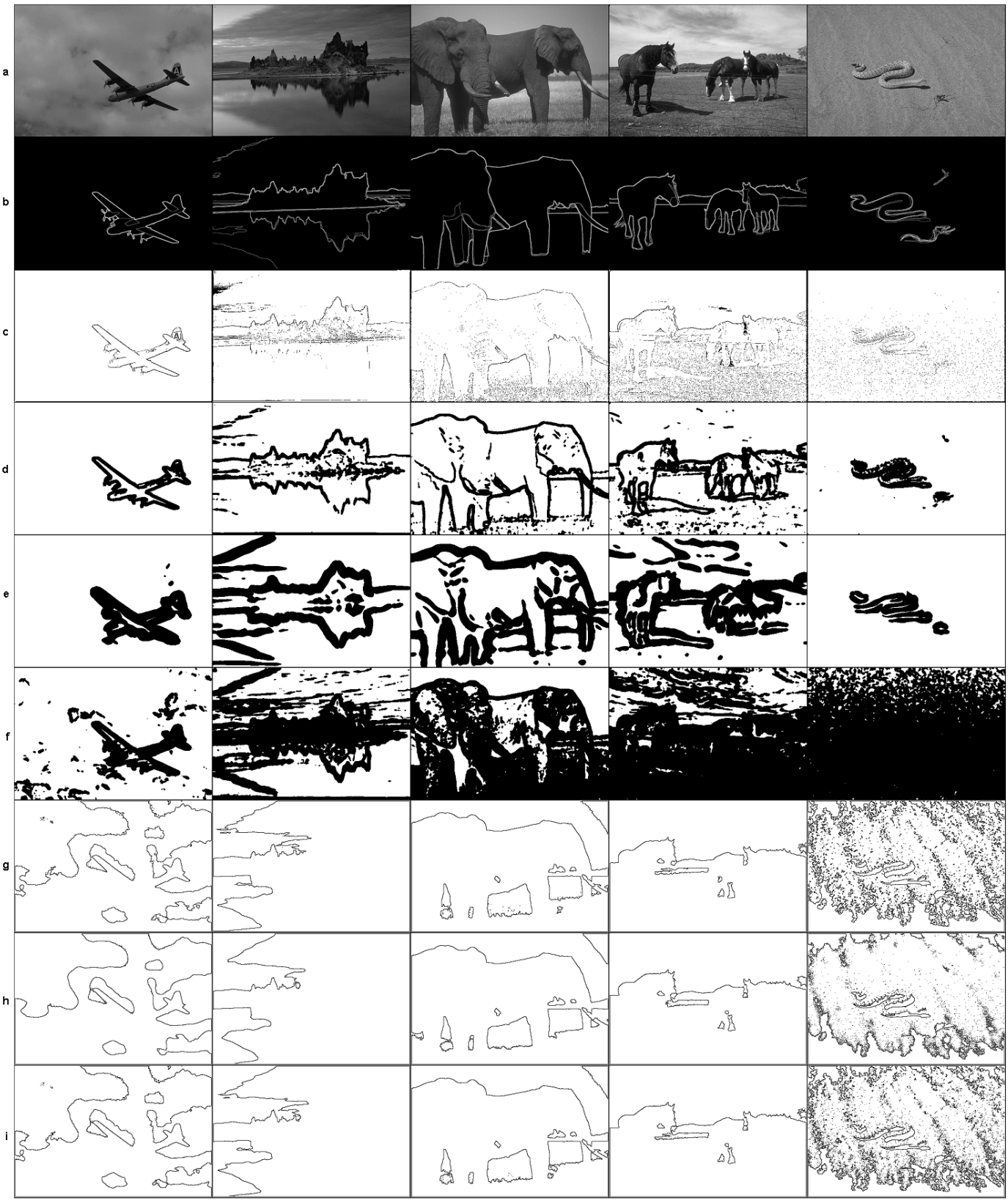

Fig.19. Comparison of results of algorithms, a) original images, b) ground truth images c) CNN-FHN model, d) Proposed algorithm using **CDiffus**, e) Proposed algorithm using **PM** diffusion model, f) Proposed algorithm using **N** diffusion model, g) Rekeczky's algorithm using **CDiffus**, h) Rekeczky's algorithm using **PM** diffusion model, i) Rekeczky's algorithm using **N** diffusion model.

REFERENCES

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," IEEE Transactions on circuits and systems, vol. 35, no. 10, pp. 1257–1272, 1988.

[2] L. O. Chua and L. Yang, "Cellular neural networks: Applications," IEEE Transactions on circuits and systems, vol. 35, no. 10, pp. 1273–1290, 1988.

[3] T. Roska and L. O. Chua, "The CNN universal machine: an analogic array computer," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, no. 3, pp. 163–173, 1993.

[4] L. O. Chua and T. Roska, "The CNN paradigm," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 40, no. 3, pp. 147–156, 1993.

[5] L. O. Chua and T. Roska, Cellular neural networks and visual computing: foundation and applications. Cambridge Univ Pr, 2002.

[6] T. Roska, "Circuits, computers, and beyond Boolean logic," International Journal of Circuit Theory and Applications, vol. 35, no. 5–6, pp. 485–496, 2007.

[7] T. Roska, "Computational and computer complexity of analogic cellular wave computers," 2003, pp. 12(4):539–556.

[8] T. Roska, "Analogic CNN Computing: Architectural, Implementation, and Algorithmic Advances-a Review," 1998, pp. 3–10.

[9] M. Pavone, P. Arena, L. Fortuna, M. Frasca, and L. Patané, "Climbing obstacle in bio-robots via CNN and adaptive attitude control," International Journal of Circuit Theory and Applications, vol. 34, no. 1, pp. 109–125, 2006.

[10] R. Tetzlaff, C. Niederh\öfer, and P. Fischer, "Automated detection of a preseizure state: non-linear EEG analysis in epilepsy by Cellular Nonlinear Networks and Volterra systems," International Journal of Circuit Theory and Applications, vol. 34, no. 1, pp. 89–108, 2006.

[11] G. Cserey, A. Falus, and T. Roska, "Immune response inspired spatial-temporal target detection algorithms with CNN-UM," International Journal of Circuit Theory and Applications, vol. 34, no. 1, pp. 21–47, 2006.

[12] I. Szatmári, "Object comparison using PDE-based wave metric on cellular neural networks," International Journal of Circuit Theory and Applications, vol. 34, no. 4, pp. 359–382, 2006.

[13] S. Xavier-de-Souza, J. A. . Suykens, and J. Vandewalle, "Learning of spatiotemporal behaviour in cellular neural networks," International Journal of Circuit Theory and Applications, vol. 34, no. 1, pp. 127–140, 2006.

[14] Z. Fodróczi and A. Radványi, "Computational auditory scene analysis in cellular wave computing framework," International Journal of Circuit Theory and Applications, vol. 34, no. 4, pp. 489–515, 2006.

[15] D. Bálya, B. Roska, T. Roska, and F. S. Werblin, "A CNN framework for modeling parallel processing in a mammalian retina," International Journal of Circuit Theory and Applications, vol. 30, no. 2–3, pp. 363–393, 2002.

[16] D. Bálya, I. Petrás, T. Roska, R. Carmona, and A. Rodriguez-Vasquez, "Implementing the multilayer retinal model on the complex-cell CNN-UM chip prototype," Int. J. Bifurc. Chaos, vol. 14, no. 2, pp. 427–451, 2004.

[17] V. I. Krinsky, V. N. Biktashev, and I. R. Efimov, "Autowave principles for parallel image processing," Physica D: Nonlinear Phenomena, vol. 49, no. 1–2, pp. 247–253, 1991.

[18] V. Perez-Munuzuri, V. Perez-Villar, and L. O. Chua, "Autowaves for image processing on a two-dimensional CNN array ofexcitable nonlinear circuits: flat and wrinkled labyrinths," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 40, no. 3, pp. 174–181, 1993.

[19] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, "Simulating nonlinear waves and partial differential equations via CNN-Part I. Basic techniques," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 42, no. 10, pp. 807–815, 1995.

[20] T. Kozek, L. O. Chua, T. Roska, D. Wolf, R. Tetzlaff, F. Puffer, and K. Lotz, "Simulating nonlinear waves and partial differential equations via CNN-Part II: Typical examples," IEEE Transactions on Circuits and Systems–I: Fundamental theory and applications, vol. 42, no. 10, pp. 807–815, 1995.

[21] C. Rekeczky, "CNN architectures for constrained diffusion based locally adaptive image processing," International Journal of Circuit Theory and Applications, vol. 30, no. 2–3, pp. 313–348, 2002.

[22] C. Rekeczky and L. O. Chua, "Computing with front propagation: active contour and skeleton models in continuous-time CNN," The Journal of VLSI Signal Processing, vol. 23, no. 2, pp. 373–402, 1999.

[23] M. Ebihara, H. Mahara, T. Sakurai, A. Nomura, A. Osa, and H. Miike, "Segmentation and edge detection of noisy image and low contrast image based on a reaction-diffusion model," The Journal of the Institute of Image Electronics Engineers of Japan, vol. 32, no. 4; ISSU 165, pp. 378–385, 2003.

[24] N. Kurata, H. Kitahata, H. Mahara, A. Nomura, H. Miike, and T. Sakurai, "Stationary pattern formation in a discrete excitable system with strong inhibitory coupling," Physical Review E, vol. 79, no. 5, p. 056203, 2009.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," 2001.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

[27] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using brightness and texture," 2002, pp. 1255–1262.

[28] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 5, pp. 530–549, 2004.

[29] T. Roska, "Cellular wave computers for brain-like spatial-temporal sensory computing," IEEE Circuits and Systems Magazine, vol. 5, no. 2, pp. 5–19, 2005.

[30] C. Rekeczky, I. Szatmari, P. Foldesy, and T. Roska, "Analogic cellular PDE machines," 2002.

[31] C. Rekeczky and T. Roska, "Calculating local and global PDEs by analogic diffusion and wave algorithms," 2001, vol. 2, pp. 17–20.

[32] L. Kék, K. Karacs, and T. Roska, "Cellular Wave Computing Library: Templates, Algorithms, and Programs," MTA-SZTAKI, Budapest, version, vol. 2.

[33] C. Rekeczky, T. Roska, and A. Ushida, "CNN-based difference-controlled adaptive nonlinear image filters," International Journal of Circuit Theory and Applications, vol. 26, no. 4, pp. 375–423, 1998.

[34] C. Rekeczky, "MATCNN: analogic CNN simuation toolbox for MATLAB," Hungarian Academy of Science, Hungary, http://lab. analogic. sztaki. hu/MATCNN, 2005.

[35] J. Canny, "A computational approach to edge detection," Readings in computer vision: issues, problems, principles, and paradigms, vol. 184, pp. 87–116, 1987.

[36] A. Nomura, M. Ichikawa, R. H. Sianipar, and H. Miike, "Edge detection with reaction-diffusion equations having a local average threshold," Pattern Recognition and Image Analysis, vol. 18, no. 2, pp. 289–299, 2008.

[37] A. Nomura, M. Ichikawa, K. Okada, H. Miike, and T. Sakurai, "Edge Detection Algorithm Inspired by Pattern Formation Processes of Reaction-Diffusion Systems," International Journal of Circuits, Systems and Signal Processing, vol. 5, No 2, 2011.

[38] P. H. Long and P. T. Cat, "Real-time Image Processing by Cellular Neural Network Using Reaction-Diffusion Model," in Knowledge and Systems Engineering, 2009. KSE'09. International Conference on, 2009, pp. 93–99.

[39] K. R. Crounse and L. O. Chua, "Methods for image processing and pattern formation in cellular neural networks: A tutorial," IEEE Transactions on Circuits and Systems I Fundamental Theory and Applications, vol. 42, no. 10, pp. 583–601, 1995.