Mansoura Journal of Computers and Information Sciences

Cellular Neural Networks Templates Learning Approach Based on Mutual Information and Firefly Algorithm for X-Ray Images de-noising

Ahmed I. Sharaf
Faculty of Computers & Information Systems, Dept. of Computer Science, Mansoura University, Egypt
Email: ahmed.sharaf.84@gmail.com

Mohamed E. Abu El-Soud
Faculty of Computers & Information Systems, Dept. of Computer Science, Mansoura University, Egypt
Email: mohamed hossieny@yahoo.com

Ibrahim M. El-Henawy
Faculty of Computers & Information Systems, Dept. of Computer Science, Zagazig University, Egypt
Email: henawy2000@yahoo.com

ABSTRACT

The Cellular Neural Network is a 2D array of analog processors which forms a parallel computing framework. The main key factors in this model are the values of the neighborhood of each cell, which are called templates. These templates are usually set by a domain expert in this framework to determine the optimal values of the templates. In this paper, a novel approach was proposed to discover the templates of the cellular neural networks based on mutual information and firefly optimization. The mutual information discovers the hidden pattern in the templates by measuring the similarities among cells. The firefly algorithm navigates the search space to find the optimal values of the templates. The benchmarking and validation have been performed on the ChestX-ray8, which is a real-world X-ray images dataset. The proposed method achieved significant results when compared to other meta-heuristics algorithms such as Genetic Algorithm and Particle Swarm Optimization.

Keywords

Cellular Neural Networks, Medical Images, X-Ray Images, Firefly Algorithm, Evolutionary Algorithms

1. Introduction

Cellular neural network (CeNN) is a modern hardware-based information-processing system which is based on cellular automata and neural network. The CeNN consists of a two-dimensional array of analog processors, where each processor is called a cell. Each cell is connected to its surrounding cells by a set of parameters named cloning template. The behavior of the CeNN is strongly dependent on the cloning template. Because of the similarity between the CeNN and neural networks, the feedback behavior is determined by a set of parameters obtained from the feedback of the neighbor cells, which is called a feedback template. The main advantages of this paradigm are robustness, high-speed, and parallel computing. Thus, it has been used in many applications related to image processing, data encryption, and forecasting problems [1]–[7]. Image processing applications could be implemented effectively on CeNN platform because of its robust architecture and high-performance [1], [4], [8]–[11]. One limitation of this platform is the challenge in building both of the cloning and feedback templates. The behavior of the CeNN is wholly depended on the values of these templates. Thus, the robustness of these templates controls the stability of the application. A common approach for designing cloning and feedback templates is based on setting their values manually. However, this approach is considered unsatisfactory because the desired behavior has to be transformed into a set of dynamical rules manually, which is a time-consuming and a painstaking process [12], [13]. A different approach used to create the CeNN templates which are based on the usage of the supervised machine learning algorithm. These algorithms are regularly based on decreasing the error utilizing a gradient descent or evolutionary algorithms.

X-ray images acquired by the direct digital radiography (DDR) are usually sensitive to various kinds of noise, such as impulse noise because of the produced error of the sensors and the communication channels. Moreover, X-ray images collected with low doses are usually contaminated by Poisson noise. These images appear as non-uniform, low-luminescence, and low-contrast because of the real-world conditions of the capturing process. Since raw images may cause malfunction diagnosis, therefore, noise cancellation and contrast enhancement are required [14]–[17].

In this paper, a hybrid algorithm was proposed based on mutual information and firefly optimization to improve the CeNN's templates learning for denoising the X-ray images. A modified based on mutual information was used to calculate the uncertainty between the features and the decision and among the features themselves. The firefly algorithm was used to find the optimal templates to remove the noise in the X-rays images. The attractiveness function of the firefly population was based on the modified mutual information measure. The hybrid algorithm used SA to generate offspring candidates when required instead of random generation. A real-world dataset consists of X-ray's images of belongs to Patients suffering from Pneumonia [18]. The obtained results were compared to traditional filters and proved the efficiency of the proposed algorithm. The remaining of this paper is organized as follows: in section (2) a brief related work was discussed to represents the state-of-art, in section (3), the fundamental knowledge of the CeNN and the firefly algorithm were presented. In section (4), the hybrid algorithm was introduced to learn the templates of noise reduction. Section (5) demonstrated the experimental results and simulation details, followed by a discussion and a comparative study. Finally, the conclusion is presented in section (6).

2. Related Work

Several studies have been proposed based on the gradient descent method such as back-propagation and recurrent learning. The back-propagation method was used to learn the templates according to its specific task exhibiting propagation solutions by decomposition of the feedback templates into both symmetric and anti-symmetric sections. However, this method is not guaranteed to find the global minimum of the function [19]. Thus, the gradient descent method can be trapped into a local minimum easily [20].Several attempts have been proposed by researchers to overcome this limitation such as genetic algorithms, and statistical optimization techniques similar to Simulated Annealing (SA) for learning CeNN templates to find the optimal values of the models [21]. A Genetic Algorithm based on the rough set model was proposed to reduce the relations among the cells and hence improves the learning process [22]. This model converted the learning process to a set of inequalities. By solving these inequalities, a collection of parameters was obtained to learn the templates. However, this methodology neglected the effect of the irrelevant features among the cells. Moreover, the feature selection criteria used in this model could quickly eliminate a vital feature when it is located among relevant features because it is not relevant to the group. Although many learning algorithms were presented to solve this problem, there is no generic solution formulated yet [23].

3. Preliminary knowledge

3.1. Cellular Neural Network (CeNN)

The majority of neural networks were classified into two categories [24]: (1) Memory-less networks. (2) Dynamical networks. Both of the Hopfield Networks and CeNN were designed as dynamical systems such that their inputs were considered as constant values and hence the initial condition controls the equilibrium point and the stability conditions. The CeNN consists of large-scale nonlinear analog processors that process the signal in a real-time and with a high speed of processing. The essential unit of this paradigm is named a cell, and these cells interact with each other directly only through their nearest neighbors, as shown in Figure (1). The neighbor's cells can communicate directly with each other. Moreover, each cell could be affected by its indirect neighbors' cells because of the propagation effect obtained from the real-time dynamics [25]. Each cell is interconnected to a set of finite neighbors that belongs to the sphere of influence r denoted by N_r and formulated as follows:

$$N_{r(i,j)} = C(k,l) \mid \max(|\{k-1\}|, |\{l-j\}|) \leq r \qquad (1)$$

Each cell $C(i,j)$ could be implemented by an equivalent RLC circuit [24] where the input is denoted by u_{ij}, a dynamical state that evolved by time is denoted by x_{ij}, and the output denoted by y_{ij}. The term $A(i,j;k,l)$ and $B(i,j;k,l)$ represents the feedback and control cloning (feed-forward) template respectively. A dynamic rule generation could be implemented due to the random effect of the feed-forward and feedback templates. The dynamical behavior of each cell is determined by a set of parameters that manage the cell interconnectivity called templates. The following equations characterize these templates:

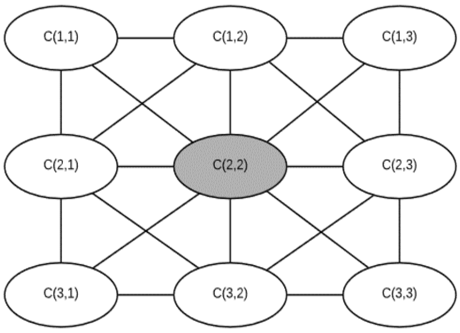


Figure 1: The structure diagram of a $2D$ CeNN of size 3×3 with the interconnection between cell 2×2 and its neighborhood $r = 1$

$$c\frac{d}{dt}X_{ij}(t) = -\frac{X_{ij}(t)}{R} + \sum_{C(k,l)\in N_r(i,j)} A_{(ij,kl)}\, Y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} B_{(ij,kl)}\, U_{kl}(t) + z \qquad (2)$$

$$Y_{ij}(t) = 0.5\big(|X_{ij}\,(t)+1| - |X_{ij}\,(t)-1|\big) \qquad (3)$$

$$-1 \leq (0) \leq 1,\ -1 \leq u_{(ij\,(t))} \leq 1, |z| \leq z_{max}, 1 \leq i \leq M, 1 \leq j \leq N \qquad (4)$$

In the CeNN system, (A, B, z) represents the local weights of the connection between each cell $C(i,j)$ and its neighbors. An analog processor could implement each cell of the CeNN system, and each cell is interconnected locally to its neighbors by matrix A and receives the feedback by the matrix B. The variable z denotes the search bias threshold. Technically, the CeNN can perform any complex task or any sophisticated algorithm by providing a memory unit and transfer registers for each cell. Thus, it improves the ability of each cell by allowing the output of each operation to be combined with the input of the next operation. For any input pattern U, the output of each cell is determined by a small set of neighbor cells as shown in Figure(1) where $r = 1$ exposed to $(2r+1) \times (2r+1)$ transparent window centered at $C(i,j)$. The output $y_{ij}(\infty)$ is considered to be a function in $(2r+1)$ of the input variables in addition to the initial state x_0, according to the complete stability theorem of the uncoupled CeNN.

$$Y_{ij} = f\big(x_0, u_1, u_2, \ldots, u_{(2r+1)^2}\big) \qquad (5)$$

The functionality of the uncoupled CeNN is a one to one mapping from U to Y for a predefined initial state $x_o$that describe the dynamic behavior at $t = 0$. The dynamic behavior of the CeNN could be implemented as a Knowledge Representation Systems such that $S = (U, x\,0 \cup c \cup Y)$ where U denotes the search space of the input patterns, C denotes the neighbor cells and Y represents the output. Thus, each row h is the knowledge systems S is considered to be a set of conditional rules. This knowledge system could be also described as CY decision rule $\Phi \rightarrow \Psi$, where Φ is a conjunction of $(2r+1)\times(2r+1)+1$ of the input cells $Ci, u\_\{hi\}$ and Ψ, denotes the classified output. Moreover, the knowledge system S represents a collection of CY decision rules, where these rules are formulated as follows:

$$Dec(C,Y) = \phi_k \rightarrow \psi_k{}_{k=1}^{m},\ 2 \leq m \leq |U| \qquad (6)$$

2.2. **Firefly Algorithm**

The firefly algorithm (FA) is a population-based stochastic search technique where each firefly presents a candidate solution in the search space [26]. Each candidate in the population moves towards the most attractive firefly which is called the leader. The leader firefly guides the other candidates to find the optimal area in the search space. The attractiveness of each firefly is determined based on the emitted light intensity. The attractiveness between two fireflies is formulated between two fireflies as follows:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \qquad (7)$$

$$r_{ij} = \sqrt{\sum_{d=1}^{D}\left(f_{id} - f_{jd}\right)^2} \qquad (8)$$

Where D represents the dimension of the problem such that $D = 1,2,\ldots r_{ij}$ is the distance between f_i and f_j. The variable β_0 represents the attractiveness at $r = 0$, γ represents the light absorption coefficient such that $\gamma \in [0,1]$. Each firefly F_i is compared with the other fireflies f_j where $j \in 1,2,\ldots N$ such that $i \neq j$ and N represents the total of the populations. If the firefly f_i is better (brighter) than f_j, then the firefly f_j moves towards the f_i with a step movement formulated as follows:

$$f_{id}(t+1) = f_{id}(t) + \beta_0 e_{ij}^{-\gamma r_{ij}^2}\left(f_{jd}(t) - f_{id}(t)\right) + \alpha\epsilon_i \qquad (9)$$

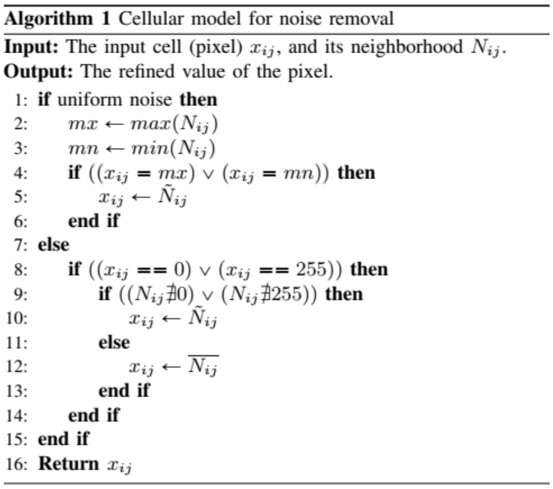
Where i denotes a uniform distributed variable such that $i \in [-0.5,0.5]$and α represents the step move such that $\alpha \in [0,1]$.

4. **Templates learning using mutual information and firefly optimization**

In this section, the template learning approach is proposed as shown in Algorithm(3). Most of the related studies discussed the relation among the neighborhood cells and the redundant cells. Although this approach was successful, the effect of features dependency was not mentioned. Moreover, some researches disregarded the relationship between the cells and the output decision. Therefore, the proposed system provided a framework to obtain a compact subset of cells depending on the mutual information and the interactivity among the cells.

The proposed method used supervised learning to learn the templates of the CeNN that remove the noise from the images. The training phase was implemented using the Algorithm(1) to build a decision table based on the noisy pixels of the images. This decision table has been used for the training stage to provide the CeNN the ability to detect the original and the noisy pixel. The CeNN is used as a supervised learning to learn the templates for the noise removal by Algorithm(1) Then, the significance of each cell is determined to remove the redundant cell in the neighborhood of the cell, as shown in Algorithm(2). Then, the firefly population is initialized randomly to create the search space of each possible solution of the templates. The termination condition of the firefly population is defined as a maximum number of iterations or when the fitness between the best and the worst solution is less 10^{-3}. when the firefly population obtains the final templates s_{best} then, the templates are transferred to the CeNN to produce the refined image. When each cell of the CeNN reaches the stable region as defined in (12) or when the difference between the mean square error of the input and the output images the less than a specific threshold θ_c. The main stages of the template learning algorithm are presented as follows:

1) Noise removal using a cellular model.
2) Neighborhood cells reduction.
3) Templates reduction using firefly optimization.



Algorithm 1 Cellular model for noise removal

```
Input: The input cell (pixel) $x_{ij}$, and its neighborhood $N_{ij}$.
Output: The refined value of the pixel.
if uniform noise then
    $mx \leftarrow max(N_{ij})$
    $mn \leftarrow min(N_{ij})$
    if $((x_{ij} = mx) \vee (x_{ij} = mn))$ then
        $x_{ij} \leftarrow \tilde{N}_{ij}$
    end if
else
    if $((x_{ij} == 0) \vee (x_{ij} == 255))$ then
        if $((N_{ij} \nexists 0) \vee (N_{ij} \nexists 255))$ then
            $x_{ij} \leftarrow \tilde{N}_{ij}$
        else
            $x_{ij} \leftarrow \overline{N_{ij}}$
        end if
    end if
end if
Return $x_{ij}$
```

4.1. Noise removal using cellular model

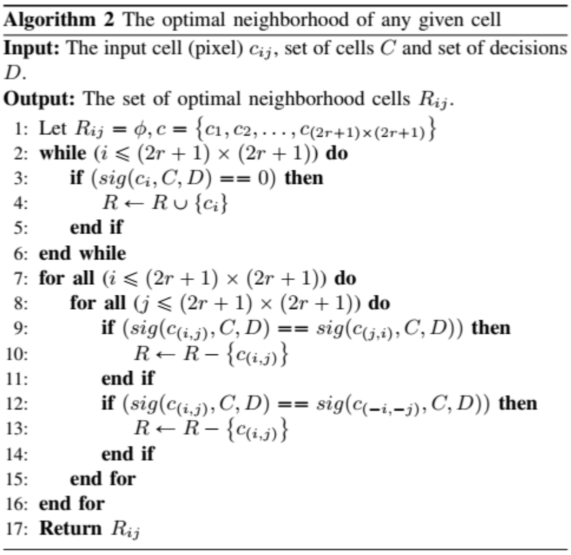
The cellular model begins with determining the nature of the existing noise in the image by calculating the most frequent values in the histogram of the given image as shown in Algorithm (1). If the most common values are black and white, then the image contains a salt and pepper noise. Otherwise, the image includes a Poisson noise. At the beginning of the cellular model, the noise type is checked if the noise is declared as uniform noise then the minimum and the maximum values from the Moore neighbors are excluded [27]. Then, the median of the neighbors is computed for the remaining cells. Whereas, if the noise is salt and pepper, the current cell is checked if it is black or white, which means it may be corrupted. Then, the median of the neighbors that do not have black or white values and assign this median for the current cell. If all the neighbors have black and white values, the average of the neighbors is computed and is assigned to the current cell. Where x_{ij} denotes the pixel at location (i,j) in the given image, $N_{i,j}$ denotes the Moore neighborhood cells, $\widetilde{N_{\iota J}}$ denotes the median of the neighborhood cells and $\overline{N_{ij}}$ denotes the mean of the neighborhood cells. Then, for each cell at location (i,j) the input and the output of the neighborhood of this cell are utilized to create the decision table using the following equation:

$$c_{ij}: u_{(i-1,j-1)}, \ldots, u_{(i+1,j+1)}, y_{(i-1,j-1)}, \ldots, y_{(i+1,j+1)} \qquad (10)$$

where u_{ij}, y_{ij} denote the input and the output of cell $c_{(i,j)}$ with $r = 1$.

4.2. Neighborhood cells reduction

In this stage, a feature reduction technique is applied to remove



```
Algorithm 2 The optimal neighborhood of any given cell
Input: The input cell (pixel) $c_{ij}$, set of cells $C$ and set of decisions $D$.
Output: The set of optimal neighborhood cells $R_{ij}$.
Let $R_{ij} = \phi, c = \{c_1, c_2, \ldots, c_{(2r+1)\times(2r+1)}\}$
while ($i \leqslant (2r+1) \times (2r+1)$) do
    if ($sig(c_i, C, D) == 0$) then
        $R \leftarrow R \cup \{c_i\}$
    end if
end while
for all ($i \leqslant (2r+1) \times (2r+1)$) do
    for all ($j \leqslant (2r+1) \times (2r+1)$) do
        if ($sig(c_{(i,j)}, C, D) == sig(c_{(j,i)}, C, D)$) then
            $R \leftarrow R - \{c_{(i,j)}\}$
        end if
        if ($sig(c_{(i,j)}, C, D) == sig(c_{(-i,-j)}, C, D)$) then
            $R \leftarrow R - \{c_{(i,j)}\}$
        end if
    end for
end for
Return $R_{ij}$
```

the redundant and superfluous cells from the neighborhood based on the mutual information between features and decision and among each feature[28]. Let $IS = (U, C \cup D)$ is an information system formulated using (10), C denotes the set of conditions and D denotes the set of decision. The significant of any condition $c_i \in C$ is determined using Algorithm(2).

$$sig(c_k, C, D) = \frac{\sum_{j=0}^{m} I(c_k, D_j)}{\sum_{j=0}^{m} I(c_k, D_j) + \sum_{i=0}^{n} I(c_k, c_i)} \qquad (11)$$

where the term $I(c_k, c_i)$ represents the redundancy between features c_k and c_i and the term $I(c_k, D_j)$ represents the dependency between the feature c_k and the decision D_j. The ranking function has been proved to be a monotonic function with a minimum value of zero and a maximum value of one[28]. Calculating the mutual information function using the standard method is an exponentially time complexity $O(n^2)$ which is time-consuming. Therefore, a non-parametric estimator was used based on a Local Non-uniformity Correction to minimize the time complexity of the feature selection process[29].

4.3. Templates learning using firefly optimization.

In this section, the firefly optimization algorithm was proposed to find the optimal value of the templates. The firefly was trained based on the decision table obtained from Algorithm(1) and Algorithm(2). Then, the optimization algorithm transfers the templates to the CeNN to benchmark the accuracy of the templates. The CeNN keeps running until is stability condition is reached. The stability condition of the CeNN is achieved when each cell reaches a stability region according to the activation function shown in Figure(2). This function control the output rendering of the CeNN such that when $x \geq 1\$\ or\ \$x \leq -1$, then the cell is stable with a value of white or black respectively. When the $-1 < x < -0.5$ or $0.5 < x < 1$ then the cell is stable for the gray scale level. Otherwise, the cell is stable with black. In this research, the value of the threshold α^{-1} was set to 0.5.

$$f(x) = \begin{cases} -1 & x \leq -1 \\ \dfrac{\alpha x}{\mathrm{arcta\ n}(\alpha)} + \alpha^{-1} & -1 < x < -0.5 \\ 0 & -0.5 \leq x \leq 0.5 \\ \dfrac{\alpha x}{\mathrm{arcta\ n}(\alpha)} - \alpha^{-1} & 0.5 < x < 1 \\ 1 & x \geq 1 \end{cases} \qquad (12)$$

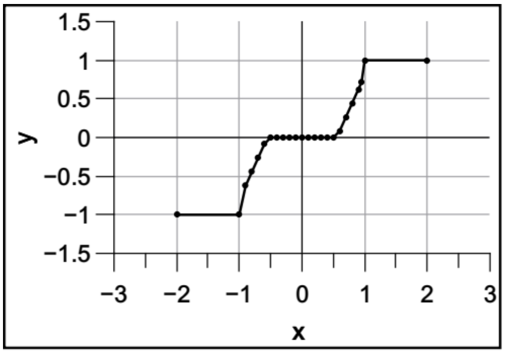


Figure 2: An illustration of the activation function of the CeNN

The initial population was generated randomly to ensure the diversity of each solution. Then, the attractiveness of each firefly was determined to select the best, the alternative, and the worst solutions. The best element was denoted by g_{best} while the worst one was denoted by g_{worst}. Furthermore, an alternative solution was denoted by S_{best} such that it should have a high rival fitness and locate in a different zone. The alternative solution was used to provide a guideline with the best one for the weak solutions and to prevent the local optima problem. A modified solution was driven from the mean of the leader and the alternative solutions to represents the optimal solution which denoted by g_{best}'. Both of the g_{best} and g_{best}' were used to guide the week solutions to move towards a better one according to the solution lightness. This algorithm is executed until it reaches the maximum number of iterations or when the difference between the fitness of the best g_{best} and the leader firefly S_{best} is less than 10^{-3}.

4.3.1. Initial Population

The firefly optimization was initialized randomly according to the optimal CeNN template structure obtained from Algorithm (2). Each firefly represents a candidate solution for the best values in the template. Then, the attractiveness of each firefly was determined to select the best firefly g_{best}, the alternative leader S_{best}, and the worst solution.

4.3.2. Search space

The training process begins with generating a decision table that consists of the input and output of each cell's neighborhood and its decision according to Algorithm(1). Then, the similarities among cells are determined using the significance measure provided in Algorithm(2) to remove the superfluous cells from the decision table. Each firefly describes a candidate configuration for the CeNN where the firefly is represented as: $\{a_1, a_2, \ldots, a_9, b_1, b_2, \ldots, b_9, z\}$ for typical CeNN with $\mathrm{r} = 1$.

4.3.3. Attractiveness function

The mean squared error (MSE) is used to measure the attractiveness between the measured image and the estimated one, as shown in (13). After computing the fitness of each candidate in the population, then the candidates were sorted in ascending order according to their fitness value. The highest element was denoted by g_{best} while the lowest one was denoted by g_{worst}. Besides, an alternative solution was denoted by S_{best} such that it should have a high rival fitness and was located in a different zone.

$$\zeta(.) = \frac{1}{MN}\sum_{i=1}^{MN}(x_i - d_i)^2 \qquad (13)$$

Where x_i denotes the measured pixel value, and MN denotes the number of pixels. The estimated pixel d_i refers to the output of the CeNN such that $d_i = y_{ij}(\infty)$ when the cellular network achieves the stability conditions defined in (2). The CeNN reaches stability regions according to the activation function defined in (12).

4.3.4. Step movement

A logistic map was used to initialize the population and thus increases the diversity and overcome the local optima. After the global solution g_{best} has been obtained, then the alternative solution S_{best} is defined with a rival fitness and a different location. Since both leaders are more likely to discover distinctive search zones, this approach minimizes the probability of being trapped in the local optima. Furthermore, the offspring solution was generated based on the global and the alternatives candidate. The obtained candidate was used to guide the low fitness candidates to move towards the optimal region and hence find the most attractive firefly. The movement step is represented by the following equations:

$$\begin{aligned} f_i(t+1) = f_i(t) + \beta_0\, C_k \left(f_j' - f_i\right) + C_k\, \varepsilon(g_{best}' - f_i) \\ + \alpha' \times sign[rand - 0.5] \end{aligned} \qquad (14)$$

$$f_j' = f_j + \sigma_1 \qquad (15)$$

$$g_{best}' = mean(g_{best} + S_{best}) + \sigma_2 \qquad (16)$$

Where $f_i(t+1)$ represents the step movement of firefly j towards the optimal region using (14) with a brighter attractiveness, and g_{best}' represents the better offspring solutions of the mean of the leader and the alternative fireflies as shown in (15). The variables σ_1 and σ_2 are two random variables set by the normal distribution. The step of the firefly was calculated, as shown in (16). Where C_k denotes the chaotic variable in the step movement and ε represents the randomized vector pre-defined in the firefly algorithm. The variable α' represents an adaptive step initialized to 0.5 to guide the diversity of the search method.

4.3.5. Diversity Method

For each repetition, the worst solution was identified after ranking as shown in Algorithm(3). Remaining candidates were led by the mean position obtained from (16) where σ denotes a random value obtained by logistic map as shown as follow:

$$f_j^{worst} = \frac{g_{best} + S_{best}}{2} + \sigma\left(f_j^{worst} \times \frac{g_{best} + S_{best}}{2}\right) \qquad (17)$$

5. Experimental Results

In this section, the experimental results were discussed to demonstrate the effectiveness of the proposed algorithm and its efficiency in finding the best CeNN templates for removing the noise in the x-rays images. The training process and the parameters of the firefly algorithm were discussed in this section. Moreover, the dataset description was discussed in this section.

5.1. Dataset Description

The ChestX-ray8 is a real-world dataset was used for the learning and benchmarking of the proposed methodology[30]. The dataset consists of $108{,}948$ frontal views X-ray images of $32{,}717$ patients with 1024×1024 resolution which is the largest collection available for public. A random noise generator was designed to simulate both the impulse and Poisson noise. The impulse noise was simulated using salt and pepper noise with 5% and 10% of noise density. The original and the contaminated images are shown in Figure(3).

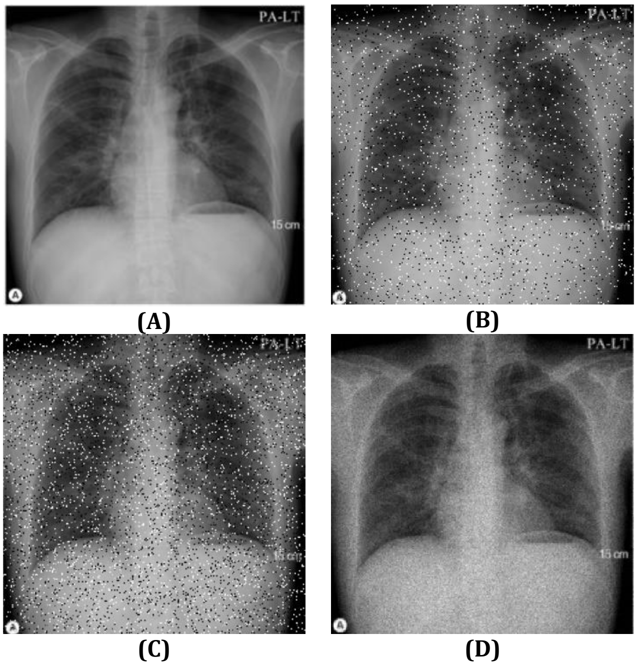


Figure 3: (A) The original image of a random patient. (B) The contaminated image with (5%) of salt and pepper noise. (C) The contaminated image with salt and pepper noise (10%). (D)The contaminated image with Poisson noise.

5.2. Simulations and Results

The development of this experiment was implemented using MATLAB on a core i7 machine with Microsoft Windows 10 operating system. The CeNN simulator was also developed using MATLAB to overcome the limitation of the hardware absence. The firefly population was initialized randomly with ten fireflies such that each candidate represents the CeNN templates. The firefly population was set with the following settings, as shown in Table(1). The contaminated gray-scale image was passed to the firefly population with the pre-defined parameters. Figure(4) demonstrates the contaminated image with 5% noise corruption on the left as the input of the CeNN, on the right Figure the denoised image after total number of iterations using the proposed methodology In Figure (5), a comparison between the 10% contaminated image and the output image were illustrated. The final CeNN template obtained by the proposed methodology is shown in (18) and (19): For the Poisson noise, the proposed algorithm produced different templates, as shown below. A comparison between both of the contaminated with Poisson noise and the denoised image was shown in Figure (6). After the feature reduction performed by the firefly, these templates could be rewritten in generic formulas as follows in (20) and (21).

$$A = \begin{bmatrix} 0.2 & 0.19 & 0.15 \\ 0.2 & 0.98 & 0.2 \\ 0.1 & 0.18 & 0.21 \end{bmatrix}, B = \begin{bmatrix} 0.1 & 0.18 & 0.15 \\ 0.13 & 0.05 & 0.14 \\ 0.11 & 0.1 & 0.13 \end{bmatrix}, z = -0.5 \tag{18}$$

Algorithm 3 Optimization algorithm of cellular neural networks templates using k-nearest attractive firefly

Input: The noisy image u, and population maximum iterations Tp_{max}
Output: The denoised image y;

```
1: Convert the image to gray-level image
2: Create the decision table using Algorithm(1)
3: Remove superfluous cells from the decision table using Algorithm(2)
4: Initialize the firefly population
5: while $(tp \leqslant Tp_{max}) \vee (\zeta(g_{best}) - \zeta(f_{worst}) \leqslant 10^{-3})$ do
6:     Calculate the fitness function $\zeta(.)$ for each firefly
7:     Choose the best and the worst solutions $f_{best}$ and $f_{worst}$
8:     Define an alternative leader as $s_{best}$, with a rival fitness and found in a different zone
9:     for all $((i \to N)$ and $(f_i \neq f_{wosrt}))$ do
10:        for all $((j = i - k \to j = i + k)$ and $(x_j \neq x_{wosrt}))$ do
11:            if $(\zeta(f_j) > \zeta(f_i))$ then
12:                Perform step movement of firefly $f_j$ towards the optimal region using (14) and (16).
13:            end if
14:        end for
15:    end for
16:    Refresh the worst solution $f_{worst}$
17:    if $\zeta(f'_{best}) > \zeta(f_{best})$ then
18:        $f_{best} \leftarrow f'_{best}$
19:    end if
20:    Rank the population and update the worst $f_{worst}$ solution with (17)
21:    $tp \leftarrow tp + 1$
22: end while
23: Obtain the best solution from the firefly population $g_{best}$
24: Assign the templates $A, B$ and $z$ to the CeNN
25: while termination conditions are not met do
26:    Set the initial state $x_0 = u$
27:    Compute the new output $y$ using the activation function (12) and (2)
28: end while
29: if $(MSE(u, y) > \theta_c)$ then
30:    return $y$
31: else
32:    goto the step 4
33: end if
34: End
```

$$A = \begin{bmatrix} 0.15 & 2.1 & 0.1 \\ 2.6 & 0.95 & 1.98 \\ 0.2 & 2.03 & 0.9 \end{bmatrix}, B = \begin{bmatrix} 0.15 & 0.25 & 0.18 \\ 0.17 & 2.6 & 0.12 \\ 0.12 & 0.18 & 0.17 \end{bmatrix}, z = 0.13 \tag{19}$$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} a & a & a \\ a & & a \\ a & a & a \end{bmatrix}, z = 0 \tag{20}$$

$$A = \begin{bmatrix} 0 & a & 0 \\ a & 1 & a \\ 0 & a & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = -0.5 \tag{21}$$

The evaluation process of the proposed system requires many measures to ensure the desired performance of the proposed systems and to ensure its effectiveness. The peak signal to noise ratio (PSNR), mean square error (MSE) and Structural

Similarity Index (SSIM) are used to determine the effectiveness of the proposed system which are defined as follows:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij} - \tilde{x}_{ij}\right)^2 \tag{22}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \tag{23}$$

$$\mathrm{SSIM} = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{24}$$

Where MN denotes the two-dimensions of the image, x_{ij} is the value of the pixel located at (i, j) in the original image and the $\tilde{x}_{ij}$ denotes the value of the pixel located at (i, j) in the reconstructed image. μ_x denotes the average of x_i, σ_x^2 is the variance of x_i, σ_{xy} denotes the co-variance of x and y. In addition, the terms C_1 and C_2 represents a constant value specified by the dynamic range of the image. Figure(7) represents a comparison among PSNR and SSIM obtained from the proposed method, PSO and GA of the contaminated image with 5% . The proposed method placed first place when compared to PSO and GA, then the PSO and finally the GA. The PSNR of the output image obtained from the proposed method increased rapidly after 800 iterations and achieved stability later. For the SSIM, the proposed method gained the first place when compared to PSO and GA. The noisy input image is compared with the generated image. When the value of the SSIM decreases, then the output image is not similar to the input. The proposed method reached stability after 800 iterations. Figure (8) represents a comparison among PSNR and SSIM obtained from the proposed method, PSO and GA of the contaminated image with 10%. The proposed method placed first place when compared to PSO and GA, then the PSO and finally the GA. The PSNR of the output image obtained from the proposed method reached a stable region after 800 iterations. For the SSIM, the proposed method gained the first place when compared to PSO and GA. The proposed method reached stability after 900 iterations. Figure (9) represents a comparison among PSNR and SSIM obtained from the proposed method, PSO and GA of the contaminated image with Poisson noise. The proposed method placed first place when compared to PSO and GA, then the PSO and finally the GA. The PSNR of the output image obtained from the proposed method reached a stable region after 720 iterations. For the SSIM, the proposed method placed first place when compared to PSO and GA. The proposed method reached stability after 900 iterations.

Table 1: Parameters settings of the firefly population

Parameter	Value
Population count $\boldsymbol{n}$	10
Light absorption $\boldsymbol{\gamma}$	0.1
Initial attractiveness $\boldsymbol{\beta_0}$	0.01
Initial adaptive step $\boldsymbol{\acute{\alpha}}$	0.5
Chaotic variable $\boldsymbol{C_k}$	0.1
$\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$	0.1
Max Iterations	1000

Table 2: Parameters settings of the PSO population

Parameter	Value
Population Size	10
Maximum Velocity $\boldsymbol{V_{max}}$	10
Acceleration coefficient $\boldsymbol{c_1}$	1.4
Acceleration coefficient $\boldsymbol{c_2}$	1.2
Initial weight $\boldsymbol{\omega}$	0.75
The Maximum Generation	1000

Table 3: Parameters settings of the GA population

Parameter	Value
Population size $\boldsymbol{n}$	10
Maximum Position $\boldsymbol{X_{max}}$	1
Number of chromosomes	6
Probability of Crossover $\boldsymbol{P_c}$	0.65
Probability of Mutation $\boldsymbol{P_M}$	0.008
The Maximum Generation	1000

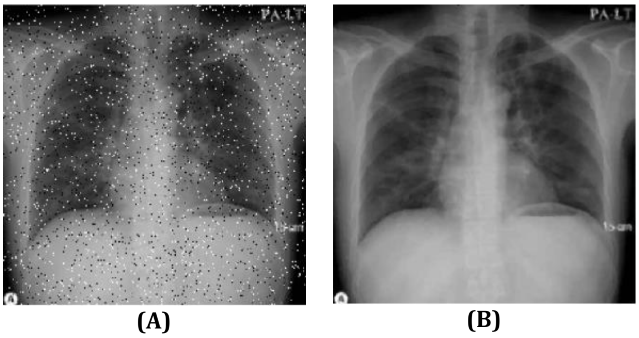


Figure 4: (A) the contaminated image with 5% noise corruption as the input of the CeNN. (B) the denoised image after the total number of iterations using the proposed methodology

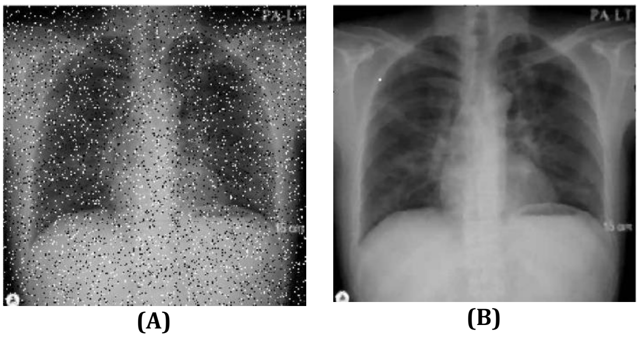


Figure 5: (A) the contaminated image with 10% noise corruption as the input of the CeNN. (B) the denoised image after the total number of iterations using the proposed methodology.

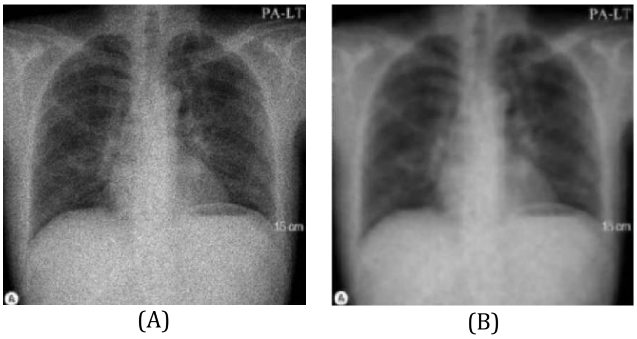


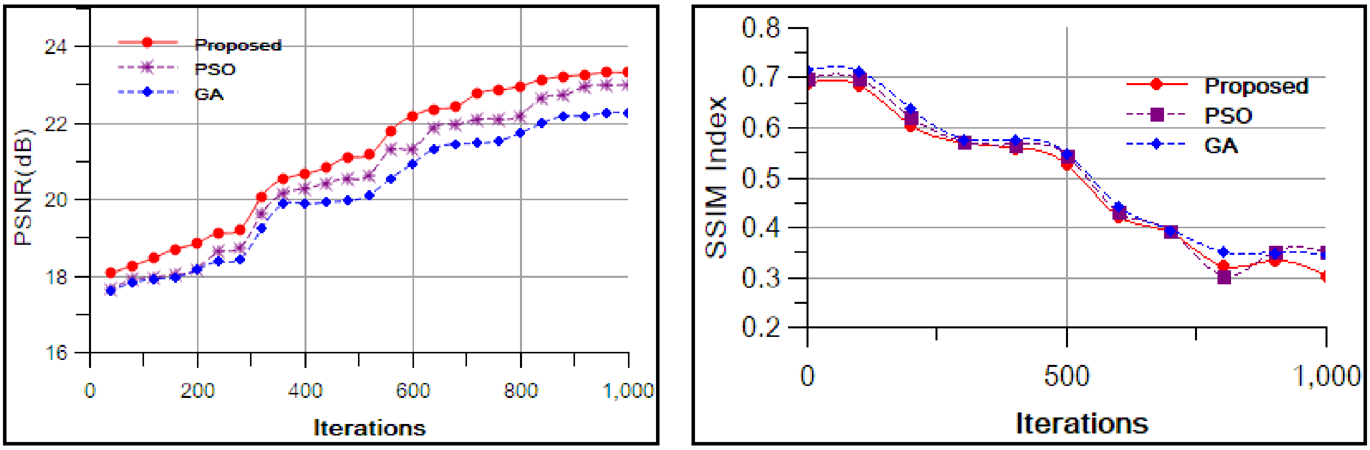
Figure 6: (A) the contaminated image Poisson noise corruption as the input of the CeNN. (B) the denoised image after the total number of iterations using the proposed methodology

5.3. Comparative Study and Discussion

In this section, a comparative study was provided to prove the efficiency of the proposed methodology. The final results were compared to Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) with pre-defined settings shown in Table(2) and Table(3) respectively. The output was measured in the total number of error pixels of the three methodologies, as shown in Figure (10). The proposed method has provided a better performance compared to both the GA and the PSO after 250 iterations. The performance of the PSO is higher than the GA with 10.25%. The proposed methodology achieved 9.6% better than the PSO. The progress of the proposed method is obtained from many factors which are summarized into the movement step and population diversity. Both of the GA and the PSO may suffer from being trapped into local optima, generate bad solutions, and inaccuracy in some problems. Therefore, a modified step movement was proposed for Firefly optimization to overcome these limitations. The results of these steps were significant for the optimization algorithm to generate new candidates and to avoid the worst solutions by keeping them into memory. The population diversity formula proved its effectiveness and efficiency when compared to the GA or the PSO from the first 100 iterations, and then the performance was in increasing slowly till the 250 iterations, then after 250 iterations and up to 650 iterations, the proposed method was the leader.

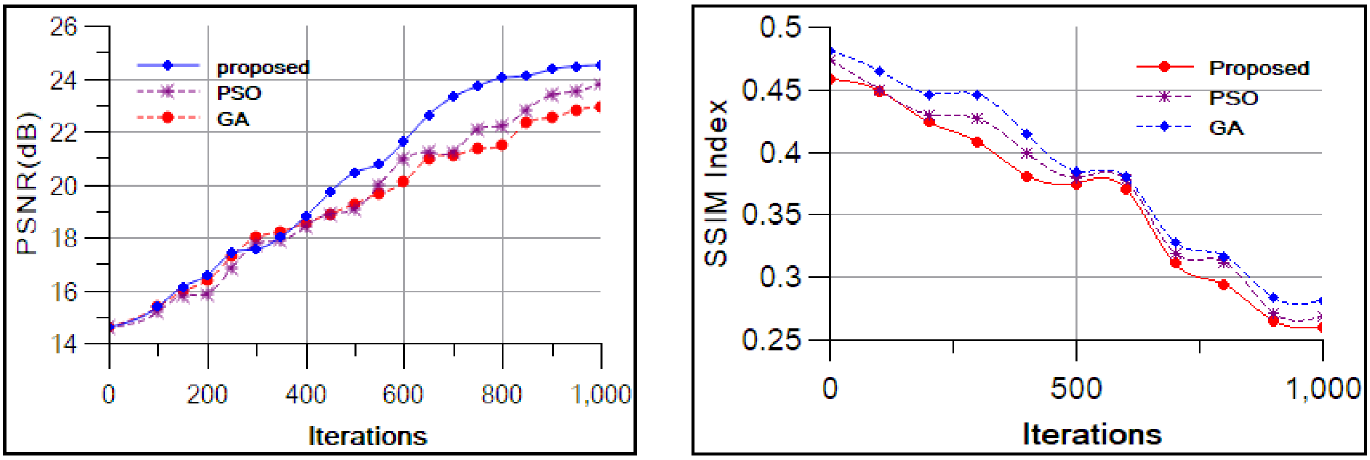
6. Conclusion

In this paper, a hybrid approach based on the modified firefly optimization was proposed to remove various kinds of noises in the medical images. The CeNN was suggested as a two-dimensional processing paradigm which ideal for handling images because of its cellular architecture. Although the CeNN is an efficient processing paradigm, the correct values of its templates represent a challenge. Thus, the proposed method provided a hybrid algorithm to solve this issue. The proposed method utilized the firefly optimization algorithm to find the optimal solutions of the templates. The attractiveness of the modified firefly was based on a mutual information uncertainty measure to specify the attractiveness of each solution. The proposed method suggested its population diversity model based on the mean location of two leader fireflies to reduce the probability of begin trapped into local optima. The ChestX-ray8 is a real-world dataset which is used for the learning and benchmarking of the proposed method. The obtained results were analyzed and discussed using the SSIM and PSNR to ensure the



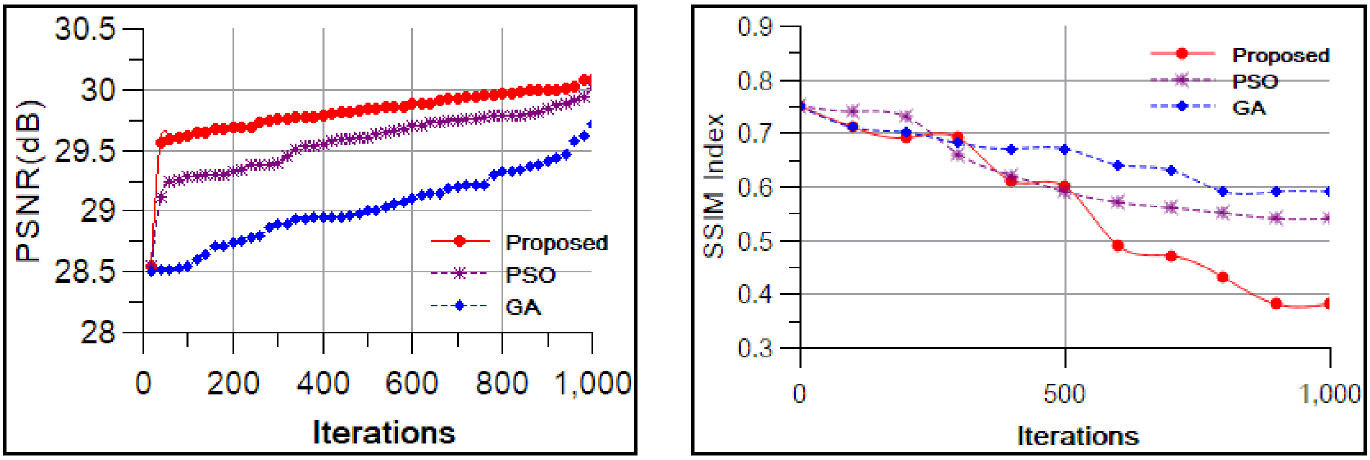
(A) (B)

Figure 7: (A) An illustration between (PSNR) and the number of iterations for the contaminated images with (5%) impulse noise obtained from the proposed method, (B) An illustration between the (SSIM) and the number of iterations for the contaminated images with (5%) impulse noise obtained from the proposed method.



(A) (B)

Figure 8: (A) An illustration between (PSNR) and the number of iterations for the contaminated images with (10%) impulse noise obtained from the proposed method, (B) An illustration between the (SSIM) and the number of iterations for the contaminated images with (10%) impulse noise obtained from the proposed method.



(A) (B)

Figure 9: (A) An illustration between (PSNR) and the number of iterations for the contaminated images with Poisson noise obtained from the proposed method, (B) An illustration between the (SSIM) and the number of iterations for the contaminated images with Poisson noise obtained from the proposed method

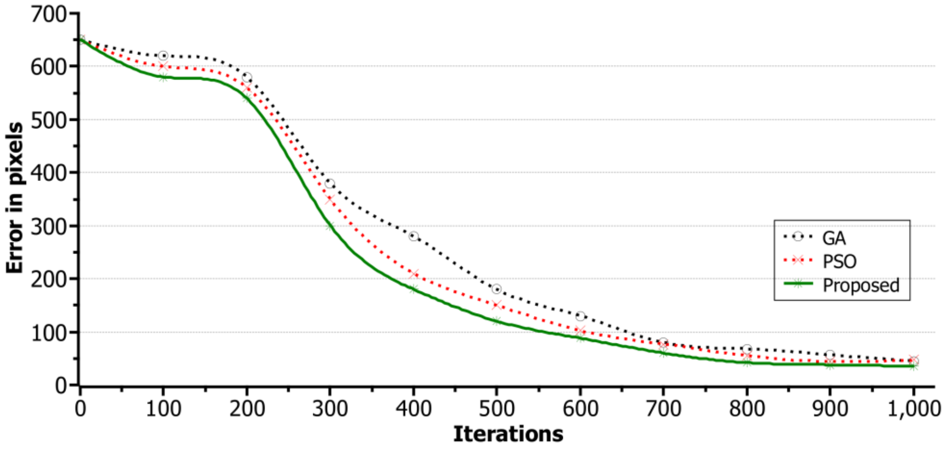


Figure 10: A comparative result between GA, PSO and the proposed method with the errors in pixels as the *y* axis and the *x* axis is the number of iterations

efficiency of the proposed method. It also provided a significant accuracy when compared to GA and PSO.

References

[1] E. Askari, S. K. Setarehdan, A. Sheikhani, M. R. Mohammadi, and M. Teshnehlab, “Modeling the connections of brain regions in children with autism using cellular neural networks and electroencephalography analysis,” *Artificial Intelligence in Medicine*, vol. 89,pp. 40–50, 2018.

[2] W. Jamrozik, “Cellular neural networks for welding arc thermograms segmentation,” *Infrared Physics & Technology*, vol. 66, pp. 18–28, 2014.

[3] M. Mozafari and R. Alizadeh, “A cellular learning automata model of investment behavior in the stock market,” *Neurocomputing*, vol. 122, pp. 470–479, 2013.

[4] K. Ratnavelu, M. Kalpana, P. Balasubramaniam, K. Wong, and P. Raveendran, “Image encryption method based on chaotic fuzzy cellular neural networks,” *Signal Processing*, vol. 140, pp. 87–96, 2017.

[5] A. Samba, Y. Busnel, A. Blanc, P. Dooze, and G. Simon, “Predicting file downloading time in cellular network: Large-scale analysis of machine learning approaches,” *Computer Networks*, 2018.

[6] S. O. Starkov and Y. N. Lavrenkov, “Prediction of the moderator temperature field in a heavy water reactor based on a cellular neural network,” *Nuclear Energy and Technology*, vol. 3, no. 2, pp. 133–140, 2017.

[7] a. Ulk ¨ u Alver, B. Cuma, and N. U. Osman, “Application of cellular¨ neural network (cnn) to the prediction of missing air pollutant data,” *Atmospheric Research*, vol. 101, no. 1–2, pp. 314 – 326, 2011.

[8] O. S. T. Adepoju and A. Ibrahim, “MEDICAL IMAGE SEGMENTATION (MIS) USING CELLULAR NEURAL NETWORK (CNN),”*International Journal of Advance Research*, vol. 1, no. 4, pp. 9–24,2013.

[9] L. Huaqing, L. Xiaofeng, L. Chuandong, H. Hongyu, and L. Chaojie, “Edge detection of noisy images based on cellular neural networks,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 9, pp. 3746 – 3759, 2011.

[10] T.-J. Su, J.-C. Cheng, M.-Y. Huang, T.-H. Lin, and C.-W. Chen, “Applications of cellular neural networks to noise cancelation in gray images based on adaptive particle-swarm optimization,” *Circuits, Systems, and Signal Processing*, vol. 30, no. 6, pp. 1131–1148, 2011.

[11] T.-J. Su, M.-Y. Huang, C.-L. Hou, and Y.-J. Lin, “Cellular neural networks for gray image noise cancellation based on a hybrid linear matrix inequality and particle swarm optimization approach,” *Neural Processing Letters*, vol. 32, no. 2, pp. 147–165, 2010.

[12] S. P. Adhikari, H. Kim, C. Yang, and L. O. Chua, “Building cellular neural network templates with a hardware friendly learning algorithm,” *Neurocomputing*, vol. 312, pp. 276–284, 2018.

[13] B. Jung-Chao and C. Chih-Hung, “Diamond in multi-layer cellular neural networks,” *Applied Mathematics and Computation*, vol. 222, no. 0, pp. 1 – 12, 2013.

[14] A. Elangovan and T. Jeyaseelan, “Medical imaging modalities: A survey,” in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–4, 2016.

[15] D. Ganguly, S. Chakraborty, M. Balitanas, and T.-h. Kim, “Medical imaging: A review,” in *Security-Enriched Urban Computing and Smart Grid* (T.-h. Kim, A. Stoica, and R.-S. Chang, eds.), pp. 504–516, Springer Berlin Heidelberg, 2011.

[16] M. Picchio and M. H. Pampaloni, “Current status and future perspectives of pet/mri hybrid imaging,” *Clinical and Translational Imaging*, vol. 5, no. 1, pp. 79–81, 2017.

[17] A. Ravishankar, S. Anusha, H. K. Akshatha, A. Raj, S. Jahnavi, and J. Madhura, “A survey on noise reduction techniques in medical images,” in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, pp. 385–389, 2017.

[18] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, and F. Yan, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131. e9, 2018.

[19] S. Sevgen, E. Yucel, and S. Arik, “Cellular neural networks template training system using iterative annealing optimization technique on ace16k chip,” in *Neural Information Processing* (C. S. Leung, M. Lee, and J. H. Chan, eds.), pp. 460–467, Springer Berlin Heidelberg, 2009.

[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, p. 436, 2015.

[21] E. Kose and Y. M ¨ us¸tak, “A new architecture for emulating cnn with template learning on fpga,” in *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications*,pp. 1–4, 2018.

[22] E. Radwan, O. Nomir, and E. Tazaki, “A new learning method for cellular neural networks templates based on hybrid of rough sets and genetic algorithms,” *International Journal of Computer Science and Information Security*, vol. 8, no. 3, pp. 155–164, 2010.

[23] D. G. G. Sierra, K. V. A. Sogamoso, and H. E. E. Cuchango, “Integration of an adaptive cellular automaton and a cellular neural network for the impulsive noise suppression and edge detection in digital images,” in *2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI)*, IEEE, jun 2019.

[24] L. Goras¸, I. Vornicu, and P. Ungureanu, *Topics on Cellular Neural Networks*, vol. 49 of *Intelligent Systems Reference Library*, book section 4, pp. 97–141. Springer Berlin Heidelberg, 2013.

[25] W. Zhou and J. M. Zurada, “A competitive layer model for cellular neural networks,” *Neural Networks*, vol. 33, no. 0, pp. 216–227, 2012.

[26] I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm and Evolutionary Computation*, no. 0, 2013.

[27] Dalhoum, A and Al-Dhamari, I and Ortega, A and Alfonseca, M, “Enhanced cellular automata for image noise removal,” in *Proceedings of the Asian Simulation Technology Conference*, pp. 69–73, 2011.

[28] A. I. Sharaf, M. A. El-Soud, and I. El-Henawy, “A Feature Selection Algorithm based on Mutual Information using Local Non-uniformity Correction Estimator,” *International Journal of Advanced Computer Science And Applications*, vol. 8, no. 6, pp. 418–423, 2017.

[29] S. Gao, G. Ver Steeg, and A. Galstyan, “Efficient estimation of mutual information for strongly dependent variables,” in *AISTATS*, 2015.

[30] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.