A NEW ART-LMS NEURAL NETWORK FOR ADAPTIVE IMAGE RESTORATION

Tzu-Chao Lin
Department of Computer Science and Information Engineering
WuFeng University
No. 117, Sec. 2, Chiankuo Rd., Minhsiung, Chiayi County 621, Taiwan
tclin@wfu.edu.tw

Received December 2009; revised April 2010

Abstract. A neural network design – the adaptive resonance theory least mean square (ART-LMS) neural network – is proposed for the restoration of images corrupted by impulse noise. The network design is based on the concept of a counterpropagation network (CPN). The ART network automatically uses a vigilance parameter to generate the cluster layer node for the Kohonen learning algorithm in CPN. In addition, the LMS learning algorithm is used to adjust the weight vectors between the cluster layer and the output layer for the Grossberg learning algorithm in CPN. The LMS algorithm is used to obtain the optimal weight for each cluster independently and minimizes the mean square error of the filter output. Experimental results show that the proposed filter based on proposed ART-LMS outperforms many well-accepted conventional filters in terms of noise suppression and detail preservation.

Keywords: Neural network, Least mean square, Impulse noise, Median filter

1. Introduction. Digital images are often distorted by impulse noise during their acquisition and transmission. The impulse noise introduced into the images in the form of bit errors and outliers severely affects the perceptual quality of the image. The efficient removal of impulse noise from digital images is an important before the images are put through image processing operations such as image compression, image segmentation and image retrieval [1,2]. Thus, there is a strong need for image restoration. However, this is a difficult task because the developing such image restoration techniques is noise removal along with preserving the image details. Developing an effective image restoration technique has become increasingly important.

A large number of approaches have been developed for image restoration. The median filter has been extensively studied due to its ability to suppress impulse noise computationally efficiently. However, the median filter is prone to alter noise-free pixels, thereby causing a number of artifacts including streaking and edge jitter. Some modified median-based filters have been proposed to overcome these shortcomings. Among those are centered-weighted median (CWM) filter [3,4], rank-ordered mean filter, progressive switching median (PSM) filter [5] and decision-based filter [27]. Basically, the task of decision-based filter is to decide when to apply the median filter and when to keep pixels unchanged. Recently, impulse noise removal based on fuzzy logic has been attracting research effort [1,6-8]. Arakawa et al. proposed a fuzzy median (FM) filter. The output of the filter is obtained as a weighted sum of the input pixel and the output of the median filter, and the weight is set based on fuzzy rules concerning the states of the input pixel. In addition, Lin and Yu proposed the partition fuzzy median (PFM) filter [1]. Though
the FM and PFM filters have demonstrated excellent robustness in filtering images, they
lie in the relatively large number of filter parameter and high computational complexity.

Since center weighted median (CWM) filter [3] and adaptive center weighted median
(ACWM) [4] filter do not obtain satisfactory results for preserving image details while
effectively suppressing impulse noise, we propose a neural-based CWM (NCWM) filter
with an adjustable center weight. Neural network has been widely used in many ap-
lications [9,10]. Neural Network models include backpropagation (BP), self-organizing
feature map (SOFM) and counterpropagation network (CPN) models [11-14]. The CPN
was designed to provide an efficient learning algorithm for solving function approxima-
tion and forecasting problems [15,16]. The CPN consists of three layers: the input layer; the
cluster (Kohonen) layer, with competitive units that do unsupervised learning; and the
output (Grossberg) layer, which is not competitive and thus only performs supervised
learning. The CPN Kohonen layer has a gradient-based unsupervised learning algorithm.
It works best when the patterns are tightly clustered in distinct groups. However, under
such a design, a neuron’s initial weight vector may be located so far away from an input
vector that it becomes far from competitive and therefore, never learns. In the present
study, we propose a neural network called ART-LMS, for which the Kohonen layer and
Grossberg layer in CPN are replaced by the adaptive resonance theory (ART) and least
mean square (LMS) algorithm, respectively [17-20]. With the help of training vectors, the
algorithm can automatically find adaptive clusters. The total number of clusters grows
dynamically with the number of input training vectors and depends on the selected vigi-
lance parameter. That is, the net can automatically create nodes until the maximum of
the cluster layer is reached.

The adaptive weight of the proposed filter is controlled by the ART-LMS neural net-
work. According to the local feature of the filter window, each input vector is classified
into a cluster and the weight corresponding to the cluster is given. The ART-LMS neural
network is employed to obtain the optimal center weight for each cluster. The performance
of the proposed filter was tested at various noise densities and on various test images. The
effectiveness and efficiency of the results are compared with other median-based filters.
Experimental results demonstrate that the proposed filter based on the ART-LMS neu-
ral network outperforms many well-accepted conventional filters in terms of both noise
suppression and detail preservation.

The rest of this paper is organized as follows. Section 2 presents the ART-LMS neural
network architecture and the learning algorithm. Section 3 describes how ART-LMS works
to achieve image restoration. Section 4 presents the results of extensive experiments to
demonstrate that the proposed filter based on ART-LMS outperforms many other filters.
Section 5 contains the conclusion.

2. ART-LMS Neural Network.

2.1. ART-LMS architecture. Adaptive resonance theory (ART) networks are a special
type of neural network designed to control the degree of similarity of patterns placed on
the same cluster node. ART networks can develop stable and plastic clustering of arbitrary
sequences of input patterns by self-organization [21-23]. Based on the framework of the
counterpropagation network (CPN), we developed the adaptive resonance theory least
mean square (ART-LMS) network to control the weight of the proposed filter. Figure
1 shows the topology of a three-layered ART-LMS network. The first layer is the input
layer, the second layer is the competitive layer (ART layer) and the third layer is the
output node layer (LMS layer). The nodes in each layer are fully interconnected with
the nodes in the adjacent layer, as shown in Figure 1. The competitive layer determines
whether a new training pattern should be classified into a specific cluster or whether a new node should be automatically generated depending on the vigilance parameter. In contrast to conventional unsupervised learning algorithms, the proposed algorithm can automatically obtain the appropriate initial weights [24].

![Figure 1. Architecture of ART-LMS](image)

The ART-LMS network functions as follows. The first layer of nodes is solely for input. The second layer is the competitive layer, where each node competes (winner takes all) based on the given inputs. ART-LMS uses the Manhattan distance to calculate the similarity between the input and the weight vector [25]. Manhattan distance $U_j$ is the difference between the input data $x = (x_1, x_2, \ldots, x_I)$ and the weight of the $j$-th node in the competitive.

$$U_j = \|x - w_j\|_1 = \sum_{i=1}^{I} |x_i - w_{ji}|$$

where $w_{ji}$ is the weight from the $j$-th node in the competitive layer to the $i$-th node in the input layer, and $I$ is the number of input nodes. $S_j$ is defined as the similarity between $x$ and $w_j$.

$$S_j = 1 - U_j$$

The weight vector for the connection to the $j$-th node in the competitive layer is given by:

$$w_j = \{w_{j1}, w_{j2}, \ldots, w_{jI}\}$$

When all similarities $S_j$ are determined, the competitive layer nodes begin to compete. The node with the largest similarity value wins. Suppose that the $p$-th node is the winner. The output $q_j$ of the $j$-th node in the competitive layer is:

$$q_j = \begin{cases} 1.0, & j = p \\ 0.0, & j \neq p \end{cases}$$

The third layer is the output layer (LMS layer). Output node $k$ of the output layer is given by:

$$y_k = \sum_{j=1}^{M} v_{kj} q_j$$
where $M$ is the maximum number of nodes in the competitive layer and $v_{kj}$ is the weight from the $k$-th node in the output layer to the $j$-th node of the competitive layer. Finally, $y_k$, $k = 1, 2, \cdots, L$ is the $k$-th computed output, where $L$ is the number of output nodes.

2.2. ART-LMS algorithm. The ART-LMS learning procedure includes two training processes. ART-LMS simultaneously trains two weights through the three layers. Table 1 shows the pseudo code of the ART-LMS algorithm. In the ART-LMS learning algorithm, the first training vector $x$ ($w_{ji} = x_i$) and the target value $v_{kj}$ are used directly to establish the first cluster node. Then, the next input training vector is compared with the first cluster node. It is assigned to the first cluster node if its similarity is larger than the vigilance parameter. Otherwise, a new cluster node is generated. That is, ART-LMS places the input vector into the most similar cluster node. This process is repeated for all training input vectors. Ultimately, ART-LMS classifies the $X$ input patterns into $M$ clusters (in general, $M < X$).

<table>
<thead>
<tr>
<th>Table 1. ART-LMS algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Let $x$ be the input data.</td>
</tr>
<tr>
<td>Set the initial weights (first input vector and first target vector).</td>
</tr>
<tr>
<td>Set the learning rate and vigilance $\delta$.</td>
</tr>
<tr>
<td>Set the maximum epoch $T$ and cluster node $M$.</td>
</tr>
<tr>
<td>(2) While (epoch number $n \leq T$ or $AMSE &gt; \theta$)</td>
</tr>
<tr>
<td>(a) Compute similarity $S_j$ of each cluster node $S_j = 1 - |x - w_j|_1$.</td>
</tr>
<tr>
<td>(b) Decide the winner node $p$, where $p$ has the largest $S_j$.</td>
</tr>
<tr>
<td>(c) If ($S_p \geq \delta$) OR (number cluster nodes $&gt; M$)</td>
</tr>
<tr>
<td>Then update weights.</td>
</tr>
<tr>
<td>$w_{pi}(t + 1) = w_{pi}(t) + \alpha(n)(x - w_{pi}(t))$.</td>
</tr>
<tr>
<td>$v_{kp}(t + 1) = \begin{cases} v_{kp}(t) - \beta(n)</td>
</tr>
<tr>
<td>(d) If ($S_p &lt; \delta$) AND (number cluster nodes $&lt; M$)</td>
</tr>
<tr>
<td>Then add one cluster node $J$.</td>
</tr>
<tr>
<td>$w_{ji} = x_i$, $i = 1, \cdots, I$.</td>
</tr>
<tr>
<td>$v_j =$ target vector $y_i$, $i = 1, \cdots, L$.</td>
</tr>
<tr>
<td>(e) Reduce learning rate.</td>
</tr>
<tr>
<td>Increase epoch number $n$.</td>
</tr>
<tr>
<td>Compute average difference of the distortion mean square error $AMSE = \frac{|MSE(n)-MSE(n-1)|}{MSE(n)}$.</td>
</tr>
</tbody>
</table>

Each training vector and its associated target value are presented to the input layer and output layer, respectively. The nodes in the cluster layer compete (winner takes all) for the input vector to be classified. The node with the largest similarity $S_j$ is the winner and sends a signal 1 to the output layer. The ART-LMS network determines the winning node for the training vector in the competitive layer. Then, weight vector $w_{pi}$, from the winner ($p$-th node) in the competitive layer to the input layer, is updated using the following learning rule.

$$w_{pi}(t + 1) = w_{pi}(t) + \alpha(n)(x - w_{pi}(t))$$

where the learning rate $\alpha(n)$ is a function of the learning epoch $n$, such as $\alpha(n) = \alpha_0 \left(1 - \frac{n}{T}\right)$, defined with a predetermined constant $0 < \alpha_0 \leq 1.0$ and the total number of learning epochs $T$. Notably, the weight vectors to the loser nodes stay unchanged.
Only the weight in the output layer connected to the winner node is updated using the least mean square (LMS) learning algorithm. The learning rule updates the weight $v_{kp}$ from the output layer to the competitive node as follows:

$$v_{kp}(t + 1) = \begin{cases} v_{kp}(t) - \beta(n)|e(t)||x - d|, & v_{kp}(t + 1) \geq 0 \\ 0, & v_{kp}(t + 1) < 0 \end{cases}$$

where the learning rate $\beta(n)$ is a function of the learning epoch $n$, such as $\beta(n) = \beta_0 \left(1 - \frac{n}{T}\right)$, defined with a predetermined constant $0 < \beta_0 \leq 1.0$ and the total number of learning epochs $T$. The variable $x$ is the real corresponding input value, and the error $e(t)$ is the difference between the desired output $d$ and the physical output $y_{kp}$. In order to speed up the convergence, the algorithm iterates the process until the average difference of the distortion mean square error (AMSE) falls below a threshold $\theta$, given by $\frac{MSE(n) - MSE(n-1)}{MSE(n)} < \theta$, where $n$ denotes an iterative epoch.

3. Image Restoration by ART-LMS. The ART-LMS network is applied to design an image filter that achieves noise suppression and detail preservation. Since the center weighted median (CWM) filters fail to suppress noise to a satisfactory degree while preserving image details, we propose a neural-based CWM (NCWM) filter with an adjustable center weight using the ART-LMS network.

3.1. Adaptive center-weighted median filter. Let $C = \{(k_1, k_2)|1 \leq k_1 \leq H, \ 1 \leq k_2 \leq W\}$ denote the pixel coordinates of the noisy image corrupted by impulse noise, where $H$ and $W$ are the image height and width, respectively. Let $x(k)$ represent the input pixel value of the noisy image at location $k \in C$. At each location $k$, the observed filter window $w\{k\}$ whose size is $N = 2n + 1$ ($n$ is a non-negative integer) is defined in terms of the coordinates symmetrically surrounding the input pixel $x(k)$.

$$w\{k\} = \{x_f(k): f = 1, 2, \cdots, n, n + 1, \cdots, N\}$$

where the input pixel $x(k) = x_{n+1}(k)$ is the center pixel. Figure 2 shows a $3 \times 3$ filter window. The output of the CWM filter is $y_c(k) = MED(w_{c}\{k\})$, where $MED$ denotes the median operation and $c$ ($c = 1, 3, 5, \cdots, N$) denotes the center weight. Now we have:

$$w_{c}\{k\} = \{x_1(k), \cdots, x_n(k), c\Diamond x_{n+1}(k), x_{n+2}(k), \cdots, x_N(k)\}$$

where $\Diamond$ represents the repetition operation. The CWM filter outputs the median value of $2n + c$ pixel values. By definition, the output of the CWM filter is the same as that the standard median filter when $c = 1$; however, the output of the CWM filter is the original input pixel (no filtering operation) when $c = N$. The weights of non-adaptive CWM filters are decided by a compromise between noise suppression and detail preservation. However, the CWM filter uniformly processes the whole noisy image, which can lead to excessive or insufficient smoothing. To fix this lack of flexibility, several researchers have improved the CWM filter under the mean absolute error (MAE) criterion [4,26-29]. In the present work, to improve noise suppression and detail preservation, the center weight $c$ in $[0, N]$ is made adjustable within the filter window $w_{c}(k)$ using the proposed ART-LMS network.

![Figure 2. Filter window about $x(k) = x_5(k)$](image-url)
3.2. Structure of the NCWM filter. The framework of the proposed NCWM filter is illustrated in Figure 3. It is composed of three parts: an observation vector for feature extraction, an ART-LMS weight controller and a CWM filter. According to the feature extraction result for the input vector, the ART-LMS weight controller gives weight $c$ to the CWM filter. The output of the NCWM filter is obtained as follows:

$$y(k) = y_c(k) = MED(w_c\{k\}), \quad c = 0, 1, \cdots, N$$

(10)

Adaptive center weight $c$ allows the CWM filter to perform various degrees of noise suppression and image detail preservation. One NCWM filter can remove most noise, but lesser impulse noise might remain. To improve filtering performance, the noise filtering procedure is progressively applied through several iterations.

![Figure 3. Structure of NCWM filter](image)

3.3. Feature extraction. For every input $x(k)$, we take into account the local features in the filter window, such as prominent signals and possible details and edges. The following three variables are defined to generate a feature vector $E\{k\}$ as the input data of the ART-LMS weight controller.

Definition 3.1. The variable $g(k)$ denotes the absolute difference between the input $x(k)$ and the median value of $w\{k\}$ as follows:

$$g(k) = |x(k) - MED(w\{k\})|$$

(11)

A large $g(k)$ value indicates that the central pixel $x(k)$ stands out among its neighboring pixels; that is, the input $x(k)$ may be corrupted by impulse noise. If we use only the $g(k)$ value to judge whether impulse noise exists, it would be difficult to fully separate impulse noise. For example, line components and edges also show up in the form of outstanding pixel values; therefore, if $x(k)$ is located on a line or an edge, it may be mistakenly interpreted as impulse noise and be removed. Therefore, it is necessary to add other observations to improve correctness. We decided to use two extra variables, $h(k)$ and $z(k)$.

Definition 3.2. $h(k) = \frac{|x(k) - x_{c1}(k)| + |x(k) - x_{c2}(k)|}{2}$

(12)

where $|x(k) - x_{c1}(k)| \leq |x(k) - x_{c2}(k)| \leq |x(k) - x_i(k)|$, and $1 \leq i \leq 2n + 1$, where $i$ is not equal to $n + 1$, $c_1$ or $c_2$ [1,14,24].

Notably, the values of $x_{c1}(k)$ and $x_{c2}(k)$ are selected to be the two closest pixel values to $x(k)$ in the filter window $w\{k\}$. If only $g(k)$ is considered, then a line component in the filter window would be identified as noise. However, if the variable $h(k)$ is also considered, then the input $x(k)$ will not be identified as noise because $h(k)$ is small.
**Definition 3.3.**

\[ z(k) = |x(k) - w_3\{k\}| \]  

(13)

If the variable \( z(k) \) is considered, then a pixel on an edge component in the filter window \( w\{k\} \) will not be detected as noise because its \( z(k) \) value is small. In the present work, the feature vectors are given by:

\[ E\{k\} = \{g(k), h(k), z(k)\} \]  

(14)

The feature vectors \( E\{k\} \) serve as the input data set to the ART-LMS weight controller.

3.4. **ART-LMS weight controller.** Adaptive weight \( c \) allows the NCWM filter to perform various degrees of noise suppression and image detail preservation. To decide the adaptive weight \( c \), we offer the ART-LMS weight controller shown in Figure 4. Basically, the architecture of the ART-LMS weight controller is the same the ART-LMS network shown in Figure 1. However, to fit the only one weight for CWM filter, the output layer has only one node. It classifies the input feature vector \( E\{k\} \) and gives the weight \( v_j \) to its corresponding cluster according to the cluster layer and the output layer in the ART-LMS network, respectively. Finally, the ART-LMS weight controller outputs the weight \( c \).

![Figure 4. Architecture of ART-LMS weight controller](image)

Table 1 lists a pseudo-code of the ART-LMS algorithm. The optimal weight \( v_j, j = 1, 2, \ldots, M \) for the NCWM filter can be obtained by minimizing the mean square error (MSE). Here, \( M \) is the maximum cluster node in the training cluster layer. The learning rule updates the weights from the input layer to the cluster layer using Equation (6). The value of \( v_j \) can be trained independently by carrying out the LMS algorithm, which can minimize the error function with respect to the cluster \( j \) [18]. The learning rule updates the weight \( v_j \) from the output layer to the cluster layer as follows:

\[
v_j(t + 1) = \begin{cases} 
  v_j(t) - \beta(n)|e(t)||x(k) - d|, & v_j(t + 1) \geq 0 \\
  0, & v_j(t + 1) < 0 \end{cases}
\]  

(15)

The learning of ART-LMS can quickly converge toward the solution. Then, the weight \( v_j \) is used as the weight \( c \) for CWM filter shown in Figure 3.
4. **Experimental Results.** To demonstrate the effectiveness of the proposed NCWM filter based on the ART-LMS network, experiments were conducted to obtain image restoration results. An impulse noise model with noise ratio $p$ can be described as follows:

$$x(k) = \begin{cases} 
    s(k), & \text{with probability } 1 - p \\
    n(k), & \text{with probability } p
\end{cases}$$

(16)

where $s(k)$ and $n(k)$ represent the original noise-free image pixel and the noise substitution for the original pixel, respectively [30]. There are two types of impulse noise: fixed-valued and random-valued impulse. In a gray-scale image, the fixed-valued impulse, which is called salt-and-pepper noise, shows up as equal to near the maximum or minimum of the allowable dynamic range. Since this fixed-valued impulse is mathematically easy, it is not practical for practical application [31]. The experiments presented here considered the random-valued impulse which is a more general noise model. The random-valued impulse uniformly distributed over the range of $[0, 255]$ is considered in 8-bit gray-scale images. The proposed filter was tested at various impulse noise densities and on several popular images including Lena, Couple, Boat, Lake, Barbara and Goldhill. The images had a size of $512 \times 512$ pixels with pixel values ranging between 0 and 255. $3 \times 3$ filter windows were used in all the experiments. The mean squared error (MSE) was employed to measure the restoration performance quantitatively. In addition, the mean absolute error (MAE) was used as a quantitative measure to evaluate the levels of the edges and the details preserved. Smaller MSE and MAE values indicate better image restoration and better image-detail preservation, respectively.

In the ART-LMS network training process, training image ‘Couple’ corrupted by 20% impulse was used. For ART-LMS network training, vigilance $\delta = 0.97$, the initial learning rate $\alpha_0 = 0.05$, $\beta_0 = 0.01$. The network dynamically generated the cluster layer nodes in the training process. Figure 5 shows the relationship between MSE values and the number of nodes in the cluster layer of the ART-LMS network. The figure suggests that the maximum number of clusters should be set to 55. Figure 6 shows the training processes for various epoch numbers and their corresponding MSE values. As shown in Figure 6, the ART-LMS network converged after six training epochs.

Since the NCWM filter adaptively selects an optimized weight through the learning process to carry out the filtering operation for each input pixel, better noise attenuation
can be expected without excessive blurring. Figure 7 shows that the NCWM filter obtained stable results after the second iteration. Thus, we collected the results after two iterations for subsequent experiments.

For comparison, the corrupted test images were also filtered using conventional as well as recently developed filters, including the standard median (MED) filter, the center weighted median (CWM) with a center weight of 3 [3], the FM filter [6], the PSM filter [5], the PFM filter [1] and the ACWM filter [4]. The filters were tested in terms of noise removal capability and detail preservation. To assess the effectiveness of the proposed ART-LMS network, the counterpropagation network (CPN) was also tested to see how well it could help with image restoration. Table 2 shows the MSE results of removing impulse noise at 20%. The performances of PFM and ACWM are very similar for all images except the Barbara image, and they yielded satisfactory results when the parameters concerned were properly set. The results of CPN seemed to strongly depend on the initial weights. In contrast, the proposed NCWM filter exhibited superior performance for all test images.
From Table 2, with consistently better results over various images, the improvement brought by the NCWM filter is evident.

Table 2. Restoration MSE results for 20% impulse noise

<table>
<thead>
<tr>
<th>Filter</th>
<th>Lena</th>
<th>Couple</th>
<th>Boat</th>
<th>Lake</th>
<th>Barbara</th>
<th>Goldhill</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>43.69</td>
<td>99.11</td>
<td>62.89</td>
<td>106.69</td>
<td>251.27</td>
<td>69.51</td>
</tr>
<tr>
<td>CWM</td>
<td>51.86</td>
<td>93.15</td>
<td>69.23</td>
<td>107.02</td>
<td>194.40</td>
<td>71.31</td>
</tr>
<tr>
<td>FM</td>
<td>32.34</td>
<td>70.81</td>
<td>44.24</td>
<td>73.33</td>
<td>184.33</td>
<td>43.25</td>
</tr>
<tr>
<td>PSM</td>
<td>37.22</td>
<td>93.13</td>
<td>53.20</td>
<td>86.82</td>
<td>249.05</td>
<td>50.07</td>
</tr>
<tr>
<td>PFM</td>
<td>24.66</td>
<td>64.94</td>
<td>36.05</td>
<td>62.81</td>
<td>197.19</td>
<td>36.76</td>
</tr>
<tr>
<td>ACWM</td>
<td>25.26</td>
<td>61.43</td>
<td>35.31</td>
<td>61.48</td>
<td>181.11</td>
<td>36.08</td>
</tr>
<tr>
<td>CPN</td>
<td>30.84</td>
<td>84.02</td>
<td>47.41</td>
<td>82.58</td>
<td>222.43</td>
<td>50.84</td>
</tr>
<tr>
<td>NCWM</td>
<td>24.42</td>
<td>59.92</td>
<td>35.12</td>
<td>59.74</td>
<td>178.79</td>
<td>35.64</td>
</tr>
</tbody>
</table>

Detail preservation performance of the NCWM filter was also compared with those of the other filters. The resultant MAE values are shown in Table 3. As the table shows, the MAE values obtained by the NCWM filter gave are much lower than those provided by the other filters for all test images. That is, the proposed NCWM filter achieved significant improvement over other techniques. This implies that the proposed filter can effectively preserve image details. The output images for the ‘Lena’ image corrupted by 20% noise probability from all filters are shown in Figure 8 for visual evaluation of the noise removal and detail preservation performance. Clearly, the MED filter produces marked blurring of the image during removal of the impulse noise. Although the CWM and FM filters preserve more details, removal of impulse noise is poor. On the other hand, the PSM, PFM and ACWM filters outperform the former two methods, but still exhibits a blurring effect and fails to sufficiently suppress impulse noise. The NCWM filter produces a restored image with better subjective visual quality by offering more noise suppression and detail preservation.

Table 3. Restoration MAE results for 20% impulse noise

<table>
<thead>
<tr>
<th>Filter</th>
<th>Lena</th>
<th>Couple</th>
<th>Boat</th>
<th>Lake</th>
<th>Barbara</th>
<th>Goldhill</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>3.46</td>
<td>5.43</td>
<td>3.92</td>
<td>5.71</td>
<td>7.98</td>
<td>4.84</td>
</tr>
<tr>
<td>CWM</td>
<td>2.76</td>
<td>4.23</td>
<td>3.12</td>
<td>4.34</td>
<td>5.70</td>
<td>3.72</td>
</tr>
<tr>
<td>FM</td>
<td>2.43</td>
<td>3.48</td>
<td>2.73</td>
<td>3.68</td>
<td>5.64</td>
<td>3.05</td>
</tr>
<tr>
<td>PSM</td>
<td>1.99</td>
<td>3.30</td>
<td>2.29</td>
<td>3.02</td>
<td>5.47</td>
<td>2.38</td>
</tr>
<tr>
<td>PFM</td>
<td>1.94</td>
<td>2.86</td>
<td>2.24</td>
<td>3.16</td>
<td>5.43</td>
<td>2.52</td>
</tr>
<tr>
<td>ACWM</td>
<td>1.81</td>
<td>3.02</td>
<td>2.05</td>
<td>2.89</td>
<td>4.67</td>
<td>2.41</td>
</tr>
<tr>
<td>CPN</td>
<td>2.58</td>
<td>4.40</td>
<td>3.02</td>
<td>4.40</td>
<td>6.89</td>
<td>3.68</td>
</tr>
<tr>
<td>NCWM</td>
<td>1.74</td>
<td>2.98</td>
<td>2.10</td>
<td>2.79</td>
<td>4.61</td>
<td>2.35</td>
</tr>
</tbody>
</table>

5. Conclusion. A neural network for image restoration was proposed. A filter built on top of the proposed ART-LMS network was developed to preserve more image details while effectively suppressing impulse noise. ART-LMS is a self-organizing neural network based on adaptive resonance theory and counterpropagation. ART-LMS uses vigilance to dynamically create cluster layer nodes. The least mean square (LMS) algorithm is used to obtain the optimal center weight for each cluster independently. The optimal weight
Figure 8. Comparison of the output images by various filters for 20% noise probability, (a) Corrupted ‘Lena’ image filtered by (b) MED, (c) CWM, (d) FM, (e) PSM, (f) PFM, (g) ACWM, (h) CPN and (i) NCWM

of each cluster minimized the mean square error of the filter output. The experimental results demonstrate that the proposed NCWM filter outperforms many conventional filters in terms of noise suppression and image detail preservation.

Acknowledgment. The author acknowledges to the National Science Council, Taiwan, for their financial support under grant NSC 98-2221-E-274-008.

REFERENCES


