A Fuzzy Morphological Neural Network: Principles and Implementation

Yonggwan Won† · Bae-Ho Lee††

ABSTRACT

The main goal of this paper is to introduce a novel definition for fuzzy mathematical morphology and a neural network implementation. The generalized-mean operator plays the key role for the definition. Such definition is well suited for neural network implementation. The first stage of the shared-weight neural network has adequate architecture to perform morphological operation. The shared-weight network performs classification based on the features extracted with the fuzzy morphological operation defined in this paper. Therefore, the parameters for the fuzzy definition can be optimized using neural network learning paradigm. Learning rules for the structuring elements, degree of membership, and weighting factors are precisely described. In application to handwritten digit recognition problem, the fuzzy morphological shared-weight neural network produced the results which are comparable to the state-of-art for this problem.

1. INTRODUCTION

Mathematical morphology has been employed in various image processing tasks and pattern recognition as a feature extraction methodology. Originally, the theoretical foundations of mathematical morphology were developed for binary images which can be represented as sets [1, 2]. Binary morphology has been extended to gray-scale morphology using umbra techniques [2, 3, 4, 5, 6, 7] and the lattices theory [8, 9].
The morphological operations are non-linear, translation invariant transformations that involve probing a signal or image with compact sets, called structuring elements. Therefore, information obtained by filtering is highly dependent upon structuring elements, as emphasized by Matheron [1], indicating that the selection of structuring elements is very important. Unfortunately, systematic design of structuring elements is difficult [10, 11, 12]. Therefore, there is a need to develop a unified methodology for generating optimal or near-optimal structuring elements for feature extraction.

Fuzzy set theory [13] has been successfully used to describe phenomena and systems with imprecise linguistic terms (i.e., tall, very tall) used in everyday. A system can be either crisp or fuzzy, depending on how the membership for an item is determined. While crisp (Boolean) systems allow the membership to be either one or zero, fuzzy systems allow for the degree of membership being a number in the range of [0, 1]. In general, the performance of fuzzy versions is superior to that of the corresponding crisp version for objective function based algorithms which iteratively minimize a criterion function until a global or local minimum is reached [41]. Furthermore, fuzzy algorithms are less probable to be trapped in local minima [14]. However, it has long been a problem to determine the degree of membership systematically.

Multilayer neural networks have been widely used to solve many problems with the development of the effective error back-propagation learning rule [4, 15, 16]. This learning rule based on an iterative gradient descent method that updates the parameters (i.e., weights and biases) from initial guess to minimize the error. Shared-weight neural network [17] is a special class of multilayer neural networks. It is a heterogeneous system that performs classification based on the high-order features combined from locally extracted. This tightly coupled approach can produce a better feature set for classification than other approaches which use the isolated feature extraction process. Defining a fuzzy morphology is an on-going research. Dougherty [18] has defined a definition, but his definition did not reflect amount of difference between the input signal and the structuring element. Furthermore, the problem of optimizing the parameters is still remaining.

Main purpose of this research was to employ the neural network learning paradigm to overcome the drawbacks of the mathematical morphology and the fuzzy logic described previously. In this paper, we introduce a new definition of fuzzy mathematical morphology and learning rules for implementation of a neural network system. In Section II, we define a fuzzy morphological operations using generalized-mean operator [19, 20]. A shared-weight neural network that performs fuzzy morphological operation for feature extraction is described in Section III. In Section IV, we present some experimental results obtained from handwritten digit recognition problem. Finally, in Section V, conclusions and further works are discussed.

2. DEFINITIONS

In this section, we first briefly describe the standard definitions of two essential operations: gray-scale erosion and dilation. We then introduce generalized-mean operator which plays key roles in defining our novel fuzzy morphological erosion and dilation. Complete mathematical description for obtaining fuzzy erosion and dilation is also provided.

2.1 Gray-scale Erosion and Dilation

Erosion and dilation are fundamental operations for mathematical morphology. The theoretical foundations of mathematical morphology lie in set theory which is well-suited for binary images [1, 2]. An extension of binary operations to gray-scale operations can be achieved by two different ways: umbral transform [2, 3, 5, 6, 7, 8] and lattice theory [8, 9]. We briefly describe the definition of those operations. De-
tails of theory, other operations, properties and applications are widely available from the morphology literature [1, 2, 4, 6, 7, 8, 21, 22].

The erosion of a function $f$ by a structuring element $g$ is defined by

$$ (f \Theta g)(x) = \max \{ y : g_y + y \leq f \} \quad (1) $$

(Fig. 1) An example of gray-scale erosion.

The erosion at a point $x$ can be done by two steps: 1) move the structuring element spatially so that its origin (origin of Euclidean space) is located at $x$, and 2) find the maximum amount we can offset (push-up) the structuring element while it is beneath the signal. Obviously, $D[g_x] \subseteq D[f]$ in order to satisfy "beneath" condition, where $D$ indicates the domain. An example of gray-scale erosion is shown in Fig. 1.

Instead of finding maximum "offset" at a point $x$, we can find the "minimum difference" between $f(z)$s and $g_z(z)$'s for all $z \in D[g_x]$. This notion leads to the formulation of erosion

$$ (f \Theta g)(x) = \max \{ f(z) - g_z(z) : z \in D[g_x] \} \quad (2) $$

Note that $(f \Theta g)(x)$ is only defined at any point where $g_x \leq f$.

Gray-scale dilation can be defined in a dual manner to gray-scale erosion. Before giving the definitions, we motivate the duality principle by showing how dilation can be viewed as an erosion. Instead of translating the structuring element and finding the maximum offset while keeping the structuring element beneath the signal, we can (i) take the "reflection" of the structuring element $g$, (ii) move the reflected structuring element $g^*$ to a point $x$, and (iii) find the "minimum" offset for the reflected-translated structuring element $(g^*)_x$ to be "above" the signal. We should note that the signal is restricted to the domain of the reflected-translated structuring element $(g^*)_x$.

(Fig. 2) An example of gray-scale dilation.

Figure 2 illustrates an example of gray-scale dilation which is formalized mathematically by

$$ (f \oplus g)(x) = \min \{ y : (g^*)_x + y \geq f \} \quad (3) $$

Instead of finding the minimum "offset" at a point $x$, as we did for gray-scale erosion, we can find the "maximum difference" between $(g^*)_x(z)$s and $(g^*)_x(z)$s for all $z \in D[(g^*)_x]$. This notion leads to the formulation of dilation

$$ (f \oplus g)(x) = \min \{ f(z) - (g^*)_x(z) : z \in D[(g^*)_x] \} \quad (4) $$

2.2 Fuzzy Erosion and Dilation

The generalized mean operator is defined [19, 20] as

$$ g(x; p, w) = \left[ \sum_i w_i x_i^p \right]^{1/p} \quad (5) $$
where
\[ \sum_{i} w_i = 1, w_i > 0, 0 \leq x_i \leq 1, \text{ and } p \neq 0. \quad (6) \]

This operator has several attractive properties. For example, the mean value monotonically increases with respect to p when the w_i's are fixed [20]. Thus, by varying p from \( -\infty \) to \( +\infty \), we can obtain all values between \( \min(x_i) \) and \( \max(x_i) \). This property was used to simulate linguistic concepts such as "at least" and "at most" by choosing appropriate values for the parameter p [23, 24, 25]. The w_i's can be thought of as the relative importance factors for the different information criteria x_i's. This property was also used for redundancy detection for a hierarchical fuzzy information fusion system [24, 25, 26].

The assumptions in (6) can be kept as hard constraints. However, we use soft assumptions in our definitions in order to implement a neural network system later on. The following theorem provides the motivation.

**Theorem:** Suppose we have a finite set \( \{ x_i ; i = 1, 2, \ldots, N \} \). If \( w_i > 0 \) and \( x_i \geq 0 \) for all i, then

\[ \lim_{p \to -\infty} g(x_i; p, w_i) = \min(x_i) \quad \text{and} \quad \lim_{p \to +\infty} g(x_i; p, w_i) = \max(x_i). \]

**Proof:** For any k,
\[ g(x_i; p, w_i) = \left[ \sum_{i} w_i x_i^p \right]^{1/p} = x_k \left[ \sum_{i \neq k} w_i \left( \frac{x_i}{x_k} \right)^p + w_k \right]^{1/p} \]
\[ = x_k \left[ f(p) \right]^{1/p} \]

where

\[ f(p) = \sum_{i \neq k} w_i \left( \frac{x_i}{x_k} \right)^p + w_k. \]

Let \( x_k \) be \( \min(x_i) \). Then,
\[ \lim_{p \to -\infty} \left( \frac{x_i}{x_k} \right)^p = \lim_{p \to +\infty} \left( \frac{x_k}{x_i} \right)^p = 0, \text{ because } 0 \leq \left( \frac{x_i}{x_k} \right) < 1. \]

Thus,
\[ \lim_{p \to -\infty} f(p) = w_k. \]

Note that
\[ \lim_{p \to -\infty} \frac{\ln f(p)}{p} = \lim_{p \to +\infty} \frac{\ln w_k}{p} = 0. \]

Therefore,
\[ \lim_{p \to -\infty} \left[ f(p) \right]^{1/p} = \lim_{p \to +\infty} \exp \left\{ \frac{\ln [f(p)]}{p} \right\} = \exp(0) = 1 \]

and
\[ \lim_{p \to -\infty} g(x_i; p, w_i) = x_k = \min(x_i). \]

Let \( x_k \) be \( \max(x_i) \). Then,
\[ \lim_{p \to +\infty} \left( \frac{x_i}{x_k} \right)^p = 0, \text{ because } 0 \leq \left( \frac{x_i}{x_k} \right) < 1. \]

In the same manner, we can show that \( \lim_{p \to +\infty} g(x_i; p, w_i) = x_k = \max(x_i) \).

Q.E.D.

From this theorem, we can formalize the erosion and dilation with the generalized-mean operator. Assume, for a while, that \( f(z) - h_k(z) \geq 0 \) and \( f(z) - (m^*)_k(z) \geq 0 \) for all z. Let us use the notation \( g(x_i; p = +\infty, w_i) \) to denote \( \lim_{p \to +\infty} g(x_i; p, w_i) \). Then erosion and dilation can be represented using the generalized mean:

Erosion: \( (f \ominus g)(x) = \min \{ f(x) - h_k(x) : z \in D[h_k] \} \)
\[ = g \{ f(x) - h_k(x) : p = -\infty, w_i \} \quad (7a) \]

Dilation: \( (f \oslash g)(x) = \max \{ f(x) - (m^*)_k(x) : z \in D[m^*_k] \} \)
\[ = g \{ f(x) - (m^*)_k(x) : p = +\infty, w_i \}. \quad (7b) \]

To avoid the assumption that \( f(z) - h_k(z) \geq 0 \) and \( f(z) - (m^*)_k(z) \geq 0 \) for all z, we can use a one-to-one, increasing function \( r:[-\infty, +\infty] \to [0, +\infty] \). This modification yields modified definitions of erosion and dilation:
Erosion: \( (f \Theta_r g)(x) = g \{ r[ f(x) - h_r(x)] ; p = -\infty \}, w_i \)  
\((8a)\)

Dilation: \( (f \oplus_r g)(x) = g \{ r[ f(x) - (m^r)_r(x)] ; p = +\infty \}, w_i \).  
\((8b)\)

Note that the weighting factor \( w_i \)'s do not play a role in these definitions. These modified definitions have empirically shown that they behave similarly to ordinary ones with the unipolar sigmoid function for \( r \) \([27, 28]\).

At this point, defining fuzzy erosion and dilation is straightforward. They are formalized as

Fuzzy Erosion: \( (f \Theta_r g)(x) = g \{ r[ f(x) - h_r(x)] ; p < 0 , w_i \} \)  
\((9a)\)

Fuzzy Dilation: \( (f \oplus_r g)(x) = g \{ r[ f(x) - (m^r)_r(x)] ; p > 0 , w_i \}. \)  
\((9b)\)

Note that the output is harmonic mean if \( p = -1 \), geometric mean if \( p = 0 \), and arithmetic mean if \( p = 1 \). Also, the weighting factor \( w_i \) is optional. If the factor is involved, the definition is a "weighted" fuzzy erosion and dilation.

### 3. NEURAL NETWORK IMPLEMENTATION

In this section, we introduce a shared-weight neural network that performs classification and feature extraction simultaneously. The first stage of the shared-weight neural network has an adequate architecture to perform the morphological operations of erosion and dilation defined in (1) and (2), respectively. Feature extraction stage of this network performs our fuzzy erosion and dilation. Learning rule for the feature extraction network is also provided.

#### 3.1 Shared-Weight Neural Network

The basic idea of the shared-weight network \([17]\) is to reduce the degrees of freedom in the network for better generalization and to form high order features from local features extracted by learned convolution kernels. Fig. 3 shows the structure of the typical shared-weight neural network. This network is composed of two parts: a feature extraction network and a fully connected feedforward network. The feedforward network is a classification network. The feature extraction network can have one or more layers, and each layer can also have one or more feature maps. The layer or layers in this network perform feature extraction by linear or non-linear convolution of its input with the kernels (also called structuring elements, templates, masks, feature detectors). The convolution output is subsampled. Therefore, the sizes of the feature maps are determined by the sampling rate for the convolution over their input.

(Fig. 3) Architecture of the shared-weight neural network with a single feature extraction layer in the feature extraction network and one hidden layer for the feedforward network.

Nodes in the first feature extraction layer have a small number of identical weights, and each node corresponds to a certain position in the input pattern. Therefore, the number of free weights in a feature map in this layer is dramatically reduced to only the size of the its kernel over its input (plus the number of the nodes in the feature map if there is one bias.
per node). On the other hand, the convolution operation in this layer automatically satisfies the translation invariant property in the equations (1) and (2). Furthermore, each node extracts local information from its input.

Each feature map in a higher feature extraction layer has as many kernels as the number of the feature maps in the next lower feature extraction layer. As for the first layer, the weights for the nodes in the same feature map are identical and thus operation is translation invariant. Furthermore, the nodes in this layer combine local information coming from the feature maps in the next lower layer. Finally, the highest layer provides the input for the feedforward network.

3.2 Node Operation

Previous work approximated the ordinary erosion and dilation with the generalized-mean operator by setting the parameter p to a large positive or negative value [7, 30]. The nodes in the feature extraction network performed a novel gray-scale Hit-Miss transform which was defined as subtraction of the dilation from the erosion. In other words, the parameter p that represents the degree of membership was fixed and a node in the feature extraction network performed both approximated erosion and dilation.

In order to allow the nodes in the feature extraction network to perform a single operation (i.e., p is either positive, negative or zero), we first formalized an equation:

\[(f \oplus g)(x) = g \{ f(x) - t_x(x) \}; p, w_i \].

This equation is equivalent to (9a) if p has a negative value and (9b) if p has a positive value. Note that the structuring element \(t_x\) represents the reflected one (\(m^*\)), if p has a positive value.

3.3 Learning Rules

In equation (10), there are three set of parameters: structuring element (t), degree (p), and weighting factor (w). Among them, structuring element and weighting factor can be fixed. Furthermore, in general, the structuring element has been selected arbitrarily (i.e., zeros). However, a structuring element designed through learning process produced better performance [28], and weighting factor provides more degree of freedom.

In this section, we provide the derivation for the learning rules required to implement the learning algorithm. Here we only show explicitly the derivation of the learning rule for the feature extraction network. The derivation for the classification network is widely available from neural network literature [16, 28, 29]. Assume that each feature extraction layer has a single feature map for simplifying the formulation, and it can be easily extended for multiple feature maps. Suppose we want to update the parameters associated with node j. Let the output of the node j be

\[O_j = \text{net}_j \{ \sum_i w_j \{ r(O_i - t_j) \}^p \}^{1/n}\].

(11)

In these equations, \(O_j\) denotes the output of the node j and \(O_i\) does the input to the node j as well as the output of the node i. Therefore, \(t_j\) is a member of the structuring element that associates the node j and the node i. In other words, the first subscript indicates the node in the next higher layer. In terms of the morphological operation, the subscript j represents the location of the origin (center) of the structuring element in the input domain.

In order to take the derivative for the equation (11), we first need to take log and obtain the following equations:

\[\log(\text{net}_j) = \frac{1}{p_j} \log(\sum_i w_j \{ r(O_i - t_j) \}^p)\].

(12)

Taking the derivative on both sides of this equation with respect to \(p_j, t_j, \) and \(w_j, \) respectively, yields

\[\frac{\partial \text{net}_j}{\partial p_j} = - \frac{\text{net}_j}{p_j^2} \log(\text{net}_j^p) + \frac{1}{p_j \cdot \text{net}_j^{p-1}} \sum_i w_j^i\]

\[\frac{\partial \text{net}_j}{\partial t_j} = - \frac{\text{net}_j}{\text{net}_j^p} \log(\text{net}_j^p) + \frac{1}{p_j \cdot \text{net}_j^{p-1}} \sum_i w_j^i\]

\[\frac{\partial \text{net}_j}{\partial w_j} = \frac{1}{p_j \cdot \text{net}_j^{p-1}} \sum_i w_j^i\]
\[
\left[ r(O_i - t_{ij}) \right]^{\eta} \log \left[ r(O_i - t_{ij}) \right],
\]

\[\frac{\partial \text{net}_j}{\partial p_i} = -w_{ji} \left( \frac{r(O_i - t_{ij})}{\text{net}_j} \right)^{\eta-1} \frac{\partial r(O_i - t_{ij})}{\partial (O_i - t_{ij})}, \quad (13b)\]

and

\[\frac{\partial \text{net}_j}{\partial w_{ji}} = \frac{1}{p_i \cdot \text{net}_j^{\eta-1}} \left[ r(O_i - t_{ij}) \right]^{\eta}. \quad (13c)\]

Again taking the derivative on both sides of the equation (12) with respect to \(O_i\) produces

\[\frac{\partial \text{net}_j}{\partial O_i} = w_{ji} \left( \frac{r(O_i - t_{ij})}{\text{net}_j} \right)^{\eta-1} \frac{\partial r(O_i - t_{ij})}{\partial (O_i - t_{ij})}. \quad (14)\]

Note that the last terms of the equations in (13b) and (14) are described as

\[\frac{\partial r(O_i - t_{ij})}{\partial (O_i - t_{ij})} = r'(O_i - t_{ij}) = r(O_i - t_{ij}) \{ 1 - r(O_i - t_{ij}) \} \quad (15)\]

because we use the unipolar sigmoid function for \(r\).

Let \(d\) represent the parameters \(p_i\), \(t_{ij}\), and \(w_{ji}\). For gradient descent learning rule to reduce the error with respect to \(d\), we apply the chain rule and obtain the equation

\[\Delta d = -\eta \frac{\partial E}{\partial d} = -\eta \frac{\partial E}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial d}. \quad (16)\]

Since the feature extraction layers are considered the hidden layer in a feedforward network, the first term of this equation can be defined and written as

\[\delta_j = -\frac{\partial E}{\partial \text{net}_j} = -\frac{\partial E}{\partial O_i} = -\sum_k \left( \frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial O_i} \right) \quad (17)\]

where the node \(k\) is the one in the next higher layer. This is called delta error of the node \(j\) in the learning rule for the standard feedforward neural network. In the same manner as for deriving (17), the first two factors of (17) can be written as

\[\frac{\partial E}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial O_i} = -\frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial \text{net}_k} = -\delta_k \quad (18)\]

which is delta error of the node \(k\) which is in the next higher layer than the node \(j\). The last term of the equation (17) has two different forms, depending upon which layer the node \(k\) belongs to. For the nodes in the highest feature extraction layer, it can be written as

\[\frac{\partial \text{net}_k}{\partial O_i} = \frac{\partial (\sum_j w_{kj} O_j + \text{bias}_k)}{\partial O_i} = w_{kj} \quad (19)\]

since the node \(k\) is a ordinary node which calculates the weighted linear combination of the inputs for its net input \(\text{net}_k\). For the nodes in other feature extraction layers, from the equation (14), it can be written as

\[\frac{\partial \text{net}_k}{\partial O_i} = w_{kj} \left( \frac{r(O_i - t_{ij})}{\text{net}_k} \right)^{\eta-1} \frac{\partial r(O_i - t_{ij})}{\partial (O_i - t_{ij})} \quad (20)\]

Therefore, from the equations (16) through (20), the learning rule for the parameters(structuring element \(t\), membership value \(p\), and weighting factor \(w\)) can be obtained. In summary, the learning rule for the parameters associated with node \(j\) can be summarized by

\[\Delta d = \eta \delta_j \frac{\partial \text{net}_j}{\partial d} \quad (21a)\]

where

\[\delta_j = \sum_k \delta_k \frac{\partial \text{net}_k}{\partial O_i} \quad (21b)\]

The last factor in (21a) is given in (13) for all parameters \(p_i\), \(t_{ij}\), and \(w_{ji}\), and the last factor in (21b) is given by either (19) or (20) depending on which layer the node belongs to.

There are several implementation details for this learning algorithm. The index \(k\) is representing all the nodes whose output calculation uses the output of node \(j\). Because \(t_{ij}\)'s are identical for all \(j\)'s in the same
feature map, $t_{ij}$'s for all $j$'s should be accumulated instead of updating $t_{ij}$'s after every $j$, and then update at the end. The updating rule for this shared-weight constraint is well described in [28]. A practical problem in implementing this learning rule is the limitation on the magnitude of the parameter $p_j$. Larger range allows more accurate approximation of the ordinary Min and Max operations. However, for large numbers, the region where the gradient for the generalized-mean is non-zero is very small and the gradient is very large in that region. Thus, the training process may oscillate for large values. Note again that the generalized-mean value is equal to harmonic mean if $p = -1$, geometric mean if $p = 0$, and arithmetic mean if $p = 1$.

4. EXPERIMENTAL RESULTS

We conducted some preliminary experiments with the shared-weight neural network that described in the previous section for handwritten digit recognition problem. We collected 1000 digits for each class from the handwritten digit data base which were extracted from the USPS mail pieces [27, 28, 30]. The digits were normalized to the fixed size of $24 \times 18$ using moment normalization[31]. Some samples of handwritten digits are shown in Fig. 4.

Among the collected images, 600 digits per class were used for training and 400 for testing the network. The networks had a single feature extraction layer with twelve feature maps and thirty hidden units for the classification network. Subsampling rate of two was used. The size of the structuring elements was $5 \times 5$. We ran five experiments with different initial values for the parameters. All parameters were learned, except the weighting factors which were set to $1/n$ where $n$ was the number of the inputs.

For all experiments, the learning rate was 0.02 and the momentum was 0.9. The training process was stopped by the pre-selected maximum epoch of 100 or Root-Mean-Squared-Error(RMSE) of 0.05. However, for most our experiments, the training process was terminated by the RMSE criteria. The parameters were initialized with the random values obtained from the range $[-0.5, 0.5]$ and the magnitude of the parameter $p$ was clipped at 3. As shown in Table 1, the network produced the results which are favorably comparable to those obtained from other approaches [17, 27, 30].

| (Table 1) Results for handwritten digit recognition problem. |
|---------------------|-----|-----|-----|-----|-----|
| Train    | Exp. 1 | Exp. 2 | Exp. 3 | Exp. 4 | Exp. 5 |
| Epoch    | 31   | 29   | 28   | 27   | 29   |
| RMSE     | 0.0482 | 0.0491 | 0.0494 | 0.0489 | 0.0487 |
| Correction Rate | 99.0% | 98.8% | 98.8% | 99.0% | 98.8% |
| Test RMSE | 0.0698 | 0.0711 | 0.0714 | 0.0702 | 0.0706 |
| Correction Rate | 95.3% | 95.0% | 94.8% | 95.0% | 94.8% |

5. CONCLUSION

We have described a novel definition for fuzzy mathematical morphology. The generalized-mean operator played a key role in this work. Our new definition is well suited for determining the parameters using neural network learning paradigm. We used our novel fuzzy morphological operator in the feature extraction stage of the shared-weight neural network. Therefore, a node in this stage of the network per-
forms fuzzy morphological operation that produces
the output values between the standard erosion and
dilation. Precise description for the learning rules is
also provided.

The shared-weight neural network that performed
our fuzzy morphological operation was applied to
handwritten digit recognition problem. We have de-
monstrated good results which are comparable to the
state-of-art for this problem [17, 27, 30].

The main goal of this paper was to introduce a new
definition for fuzzy morphological erosion and
dilation. We believe that further refinements of these
definitions are necessary. Furthermore, our definitions
can be used as a node operation for other networks
such as the standard feedforward network, even
though we selected the shared-weight network for our
preliminary study. Also more applications including
gray-level images and signals should be considered for
future works.

REFERENCES


[3] Y. Nakagawa and Rosenfeld, "A Note on the
Use of local Min and Max Operation in Digital
Picture Processing," *IEEE Trans. on System, Man


cal Methods in Image and Signal Processing.*

"Grayscale Morphology." in Proc. IEEE Conf. on
Computer Vision and Pattern Recognition, Miami
FL, 1986.

filters-Part 1: Their Set-theoretic Analysis and
Relations to Linear Shift-invariant filters," *IEEE

phology, Vol. 2: Theoretical Advances*, Academic

[9] H.J.A.M. Heijmans, "Theoretical Aspects of
Gray-level Morphology." *IEEE Trans. on Pattern
Analysis and Machine Intelligence*, Vol. 13, No.6,

[10] E.R. Dougherty and R.P. Loe, "Optimal Mean-
Absolute Hit-or-Miss Filters." *Optical Engineer-

Optimal Morphological Filter Design via Struct-

[12] P.R. Loe, "Morphological Filter Mean-Absol-
ute-Error Representation Theorems and Their
Application to Optimal Morphological Filter De-
sign." Ph.D. Thesis, Rochester Institute of Tech-


[14] J.C. Bezdek, *Pattern Recognition with Fuzzy Ob-
jective Function Algorithms*. New York: Plenum,
1981.

"Learning Internal Representations by Error Prop-
agation." in D.E. Rumelhart and J.L. McCleld-
and, editors, *Parallel Distributed Processing: Ex-
plorations in the Microstructure of Cognition*,

Distributed Processing: Explorations in the Micro-

[17] Y. le Cun, L. D. Jackel, B. Boser, J. S. Denker,
H. P. Graf, I. Guyon, D. Henderson, R. E.


이 배호
1978년 한양대학교 전자공학과
(학사)
1980년 한국과학기술원 전기 및
전자공학과(석사)
1993년 University of Missouri,
Electrical and Computer Engineering(박사)
1980년~1983년 국방과학연구소 근무
1993년~현재 전남대학교 컴퓨터공학과 조교수
관심분야: 인공지능, Robotics 응용, 컴퓨터 Vision, 멀티미디어