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.

퍼지 수리 형태학적 신경망:원리 및 구현

원 용 관† · 이 배 호††

요 약

본 논문의 퍼지 수리 형태학의 새로운 정의와 신경망을 이용한 이의 구현을 소개함에 주 목적을 두고 있다. 이 새로운 정의에는 generalized-mean 연산자가 중요한 역할을 하고 있다. 본 정의는 신경망을 이용한 구현에 매우 적합하다. 연결자 공유(shared-weight) 신경망의 전반부는 수리 형태적 연산을 수행하기에 적합한 구조를 가지고 있다. 이 연결자 공유 신경망은 퍼지 수리형태학적 연산을 이용하여 추출된 특성 정보를 근거로하여 형태 분류를 수행한다. 따라서, 본 퍼지 정의의 파라미터들은 신경망의 학습 기법을 이용하여 최적화를 기할 수 있다. 구조소들(structuring elements), membership의 값, 그리고 가중 요소(weighting factor)들을 결정하기 위한 학습방법(learning rule)들이 자세히 열거되어 있다. 적용 예로서 필기체 숫자 인식 문제에 응용한 결과, 퍼지수리 형태학을 이용한 신경망은 이 문제에 있어 현존하는 최고의 결과들과 충분히 견줄만한 결과를 보여주고 있다.

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].

† 정 회 원:한국전자통신연구소 인공지능 연구실
†† 정 회 원:전남대학교 컴퓨터공학과
논문접수:1995년 11월 20일, 심사완료:1996년 5월 15일

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 deteminе 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 pradigm 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

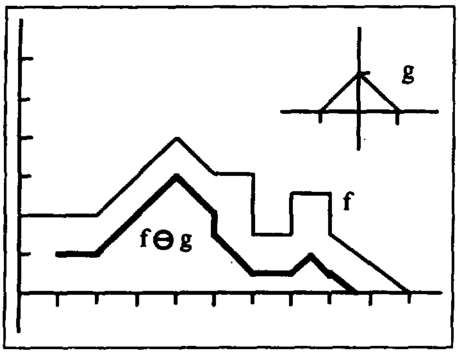
In this section, we first briefly describe the standard definitions of two essential operations; gray-scale erosion and dilation. We then introduce generalized-mean operator which plays key roles in defining our novel fuzzy morphological erosion and dilation. Complete mathematical description for obtaining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

2.1 Gray-scale Erosion and Dilation

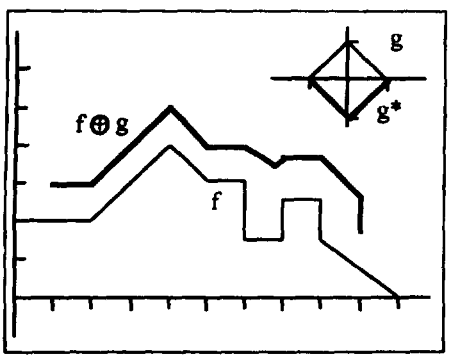
Erosion and dilation are fundamental operations for mathematical morphology. The theoretical foundations of mathematical morphology lie in *set theory* which is well-suited for binary images [1, 2]. An extension of binary operations to gray-scale operations can be achieved by two different ways: umbra transfomr [2, 3, 5, 6, 7, 8] and lattice theory [8, 9]. We briefly describe the definition of those operations. Details 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

$$(f \ominus g)(x) = \max\{y : g_x + y \ll f\}. \tag{1}$$



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



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

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 examples 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 $g_x(z)$'s for all $z \in D[g_x]$. This notion leads to the formulation of erosion

$$(f \ominus g)(x) = \max\{f(z) - g_x(z) : z \in D[g_x]\}. \tag{2}$$

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

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 $(g^*)_x$. Figure 2 illustrates an example of gray-scale dilation which is formalized mathematically by

$$(f \oplus g)(x) = \min\{y : (g^*)_x + y \gg f\}. \tag{3}$$

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

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

2.2 Fuzzy Erosion and Dilation

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

$$g(x_i; p, w_i) = \left[\sum_i w_i x_i^p\right]^{1/p} \tag{5}$$

where

$$\sum_i w_i = 1,\ w_i > 0,\ 0 \le x_i \le 1, \text{ and } p \ne 0. \tag{6}$$

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'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.

<u>*Theorem*</u>: Suppose we have a finite set $\{x_i; i = 1, 2, \cdots, N\}$. If $w_i > 0$ and $x_i \ge 0$ for all i, then

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

<u>*Proof*</u>: For any k,

$$g(x_i; p, w_i) = \left[\sum_i w_i x_i^p\right]^{1/p} = x_k\left[\sum_{i\ne k} w_i\left(\frac{x_i}{x_k}\right)^p + w_k\right]^{1/p}$$

$$= x_k[f(p)]^{1/p}$$

where

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

Let x_k be $\min\{x_i\}$. Then,

$$\lim_{p\to-\infty}\left(\frac{x_i}{x_k}\right)^p = \lim_{p\to+\infty}\left(\frac{x_k}{x_i}\right)^p = 0, \text{ because } 0 \le \left(\frac{x_k}{x_i}\right) < 1.$$

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,

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

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

Let x_k be $\max\{x_i\}$. Then,

$$\lim_{p\to+\infty}\left(\frac{x_i}{x_k}\right)^p = 0, \text{ because } 0 \le \left(\frac{x_i}{x_k}\right) < 1.$$

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

Q.E.D.

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

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

$$\text{Dilation}: (f \oplus g)(x) = \max\{f(z) - (m^*)_x(z) : z \in D[(m^*)_x]\}$$
$$= g\{f(z) - (m^*)_x(z) : p = +\infty, w_i\}. \tag{7b}$$

To avoid the assumption that $f(z)\text{-}h_x(z) \ge 0$ and $f(z)\text{-}(m^*)_x(z) \ge 0$ 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: $(f \ominus_r g)(x) = g\{r[f(z) - h_x(z)]; p = -\infty, w_i\}$ (8a)

Dilation: $(f \oplus_r g)(x) = g\{r[f(z) - (m^*)_x(z)]; p = +\infty, w_i\}$. (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: $(f \ominus_f g)(x) = g\{r[f(z) - h_x(z)]; p < 0, w_i\}$ (9a)

Fuzzy Dilation: $(f \oplus_f g)(x) = g\{r[f(z) - (m^*)_x(z)]; p > 0, w_i\}$. (9b)

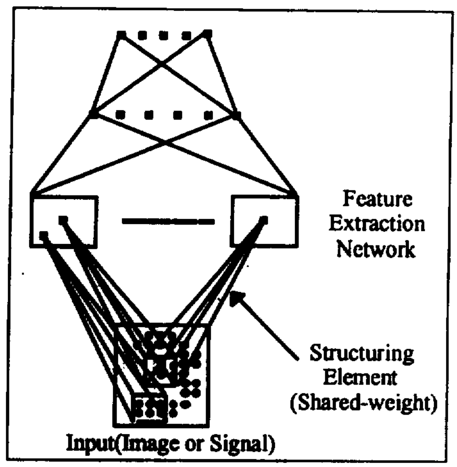
Note that the output is harmonic mean if $p = -1$, geometric mean if $p = 0$, and arithmetic mean if $p = 1$. Also, the weighting factor w_i is optional. If the factor is involved, the definition is a "weighted" fuzzy erosion and dilation.

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 dialtion 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:

$$(f \oplus_f g)(x) = g\{r[f(z) - t_x(z)]; p, w_i\}. \qquad (10)$$

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 $(m^*)_x$ 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

$$O_j = net_j\{\sum_i w_{ji}[r(O_i - t_{ji})]^{p_j}\}^{1/p_j}. \qquad (11)$$

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_{ji} 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:

$$\log(net_j) = \frac{1}{p_j}\log\{\sum_i w_{ji}[r(O_i - t_{ji})]^{p_j}\}. \qquad (12)$$

Taking the derivative on both sides of this equation with respect to p_j, t_{ji}, and w_{ji}, respectively, yields

$$\frac{\partial net_j}{\partial p_j} = -\frac{net_j}{p_j^2}\log\left(net_j^{p_j}\right) + \frac{1}{p_j \cdot net_j^{p_j-1}}\sum_i w_{ji}\cdot$$

$$\left[r(O_i-t_{ji})\right]^{p_i}\log\left[r(O_i-t_{ji})\right], \tag{13a}$$

$$\frac{\partial net_j}{\partial t_{ji}}=-w_{ji}\left\{\frac{r(O_i-t_{ji})}{net_j}\right\}^{p_i-1}\frac{\partial r(O_i-t_{ji})}{\partial(O_i-t_{ji})}, \tag{13b}$$

and

$$\frac{\partial net_j}{\partial w_{ji}}=-\frac{1}{p_j\cdot net_j^{p_i-1}}\left[r(O_i-t_{ji})\right]^{p_i}. \tag{13c}$$

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

$$\frac{\partial net_j}{\partial O_i}=w_{ji}\left\{\frac{r(O_i-t_{ji})}{net_j}\right\}^{p_i-1}\frac{\partial r(O_i-t_{ji})}{\partial(O_i-t_{ji})}. \tag{14}$$

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

$$\frac{\partial r(O_i-t_{ji})}{\partial(O_i-t_{ji})}=r'(O_i-t_{ji})=r(O_i-t_{ji})\{1-r(O_i-t_{ji})\} \tag{15}$$

because we use the unipolar sigmoid function for r.

Let d represent the parameters p_j, t_{ji}, 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

$$\Delta d=-\eta\frac{\partial E}{\partial d}=-\eta\frac{\partial E}{\partial net_j}\cdot\frac{\partial net_j}{\partial d}. \tag{16}$$

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

$$\delta_j=-\frac{\partial E}{\partial net_j}=-\frac{\partial E}{\partial O_j}=-\sum_k\left(\frac{\partial E}{\partial O_k}\cdot\frac{\partial O_k}{\partial net_k}\cdot\frac{\partial net_k}{\partial O_j}\right) \tag{17}$$

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

$$-\frac{\partial E}{\partial O_k}\cdot\frac{\partial O_k}{\partial net_k}=-\frac{\partial E}{\partial O_k}=\delta_k \tag{18}$$

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

$$\frac{\partial net_k}{\partial O_j}=\frac{\partial(\sum_{j'} w_{kj'}O_{j'}+bias_k)}{\partial O_j}=w_{kj} \tag{19}$$

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

$$\frac{\partial net_k}{\partial O_j}=w_{kj}\left\{\frac{r(O_j-t_{kj})}{net_k}\right\}^{p_k-1}\frac{\partial r(O_j-t_{kj})}{\partial(O_j-t_{kj})} \tag{20}$$

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

$$\Delta d=\eta\,\delta_j\frac{\partial net_j}{\partial d} \tag{21a}$$

where

$$\delta_j=\sum_k\delta_k\frac{\partial net_k}{\partial O_j} \tag{21b}$$

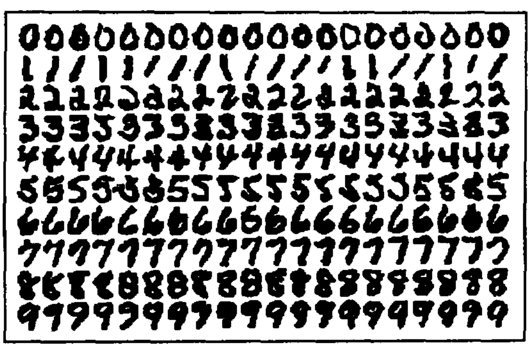
The last factor in (21a) is given in (13) for all parameters p_j, t_{ji}, and w_{ji}, and the last factor in (21b) is given by either (19) or (20) depending on which layer the node belongs to.

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_{ji}'s are identical for all j's in the same feature map, t_{ji}'s for all j's should be accumulated instead of updating t_{ji}'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 24×18 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 5×5. 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.



(Fig. 4) Some samples of size-normalized handwritten digits.

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.

		Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Train	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 performs 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 demonstrated 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

[1] G. Matheron, *Random Sets and Integral Geometry*, Wiley, New York, 1975.

[2] J. Serra, *Image Analysis and Mathematical morphology, Vol. 1*, Academic Press, New York, 1982.

[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 and Cybernetics*. Vol. 8, 1978.

[4] S.R. Sternberg, "Grayscale Morphology." *Computer Vision, Graphics, and Image Processing*, Vol. 35, pp. 333-355, 1986.

[5] C.R. Giardina and E.R. Dougherty, *Morphological Methods in Image and Signal Processing*. Englewood Cliffs, NJ, Pretice-Hall, 1988.

[6] R.M. Haralick, S.R. Stenberg and X. Zhuang, "Grayscale Morphology." in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Miami FL, 1986.

[7] P. Maragos and R.W. Schafer, "Morphological filters-Part 1 : Their Set-theoretic Analysis and Relations to Linear Shift-invariant filters," *IEEE Trans. Acoust. Speech Signal Processing*, Vol. 35, pp. 1153-1169, 1987.

[8] J. Serra, *Image Analysis and Mathematical morphology, Vol. 2 : Theoretical Advances*, Academic Press, New York, 1982.

[9] H.J.A.M. Heijmans, "Theoretical Aspects of Gray-level Morphology." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No.6, pp. 568-582, 1994.

[10] E.R. Dougherty and R.P. Loce, "Optimal Mean-Absolute Hit-or-Miss Filters." *Optical Engineering*, Vol. 32, No. 4, pp. 815-827, 1993.

[11] R.P. Loce and E.R. Dougherty, "Facilitation of Optimal Morphological Filter Design via Structuring Element Libraries and Design Constraints." Optical Engineering, Vol. 31, No. 5, pp. 1008-1025, 1992.

[12] P.R. Loce, "Morphological Filter Mean-Absolute-Error Representation Theorems and Their Application to Optimal Morphological Filter Design." Ph.D. Thesis, Rochester Institute of Technology, May, 1993.

[13] L.A. Zadeh, "Fuzzy Sets." *Informa. Contr.*, vol. 8, pp. 338-353, 1965.

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

[15] D.E. Rumelhart, G.E. Hiton and R.J. Williams, "Learning Internal Representations by Error Propagation." in D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986.

[16] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing : Explorations in the Microstructure of Cognition : Foundations*, Vol. 1, MIT Press, Cambridge, MA, 1986.

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

Howard, and W. Hubbard, "Handwritten Digit Recognition: Application of Neural Network Chips and Automatic Learning," *IEEE Communications Mgazine*, November, pp. 41-64, 1989.

[18] E. Dougherty, *Personal communication* about definition for fuzzy morphology, Manuscript was in preparation, 1994.

[19] A. Dujmovic, "Weighted Conjunctive and Disjunctive Means and Their Application in System Evaluation." *Publikacije Electrotechnickog Fakulteta Beograd, Serija Matematika i Fizika*, Vol. 483, pp. 147-158, 1974.

[20] H. Dyckhoff and W. Pedrycz, "Generalized Means as a Model of Compensation Connectives." *Fuzzy Sets and Systems*, Vol. 14, No. 2, pp. 143-154, 1984.

[21] E.R. Dougherty, *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press, 1992.

[22] P. Maragos, "A Representation Theory for Morphological Image and Signal Processing." *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 586-599, 1989.

[23] R. Krishnapuram and J. Lee, "Propagation of Uncertainty in Neural Networks." in *Proc. of SPIE Conference on Robotics and Computer Vision*, Cambridge, MA, pp. 377-383, November, 1988.

[24] R. Krishnapuram and J.Lee, "Fuzzy-Connective-Based Hierarchical Aggregation Networks for Decision Making," *Fuzzy Sets and Systems*, Vol. 46, No. 3, 1992.

[25] J. Lee, "Fuzzy-Set-Connective-Based Information Fusion Techniques and Their Application to Computer Vision." Ph.D. Dissertation, University of Missouri-Columbia, 1990.

[26] R. Krishnapuram and J. Lee, "Fuzzy-Set-Based Hierarchical Networks for Information Fusion in Computer Vision." *Neural Networks*, Vol. 5, pp. 335-350, 1992.

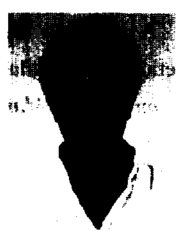
[27] P.D. Gader, Y. Won and M.A. Khabou, "Image Algebra Networks for Pattern Classification." *SPIE Mathematical Morphology and Image Algebra*, pp. 157-168, July, 1993.

[28] Y. Won, "Nonlinear Correlation Filter and Morphology Neural Networks for Image Pattern and Automatic Target Recognition." Ph.D. Dissertation, University of Missouri-Columbia, 1995.

[29] P.J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavior Science." Doctoral Dissertation, Applied Mathematics, Harvard University, Boston, MA, November 1974.

[30] P.D. Gader and Mohamed A. Khabou, "Automated Feature Generation for Handwritten Digit Recognition by Neural Networks." in *Proc. of the Int'l Workshop Frontiers Handwriting Recognition*, pp. 21-31, Buffalo, NY, 1993.

[31] R. Casey, "Moment Normalization of Handprinted Characters." in *IBM J. Res. Develop.*, pp. 548-557, September, 1970.



원 용 관

1987년 한양대학교 전자공학과 (학사)
1991년 University of Missouri, Electrical and Computer Engineering(석사)
1995년 University of Missouri, Electrical and Computer Engineering(박사)
1987년~1988년 금성통신 근무
1992년~1993년 U.S. Postal Service 지원, 필기체 문자 인식
1994년~1995년 U.S. Air Force 지원, 자동 목표물 인식
1995년 11월~현재 한국전자통신연구소, 인공지능연구실
관심분야: Image/Signal Processing, Pattern Recognition(자동목표물, 문자, 얼굴), Virtual Reality, Neural networks, Fuzzy logic

이 배 호

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