

## 신경망을 이용한 인쇄체 한자의 인식

正會員 李 成 範\* 正會員 吳 鍾 旭\*\* 正會員 南宮 在 贊\*\*\*

An recognition of printed chinese character  
using neural networkSung Bum Lee\*, Jong Wook Oh\*\*, Jae Chan Namkung\*\*\* *Regular Members*

## 요 약

본 논문에서는 종래의 결정론적 방법과 신경망을 이용하여, 인쇄체 한자를 인식하는 방법을 제안하였다. 먼저 한자를 구성하는 획성분의 4방향벡터를 추출하였다. 다음에 구해진 방향벡터에 무게중심의 메쉬를 만든 다음, 각 메쉬내의 흑화소 길이로  $8 \times 8$ 의 특징 매트릭스를 구성하였다. 정규화한 특징 매트릭스 값은 14의 문자형식으로 1차 분류하기 위해 신경망에 입력으로 하였고, 이 분류된 문자는 부수를 인식하는 부수인식 신경망에서 다시 2차분류하였다. 마지막으로 2차분류된 문자는 입력한자와 표준한자와의 유사도를 적용하여 최종인식을 행하였다. 본 알고리즘이 한자의 인식에 유효함을 보였다.

## Abstract

In this paper, we propose to method of recognizing printed chinese characters which combine the conventional deterministic methods and the neural networks. Firstly, we extract four directional vector of strokes from chinese characters. Secondly, we make the mesh of the center of gravity in the vector and then constitute the  $8 \times 8$  feature matrix using black pixel lenth from each meshes.

This normalized feature matrix value offer as the input of neural network for classifying into the 14 character types. And this calssified character classify again into Busu group by the Busu recognizing neural network. Finally, we recognize each characters using the distance of similarity between input characters and reference characters. The usefulness of the proposed algorithm is evaluated by experimenting with recognizing the chinese characters.

## I. 서 론

컴퓨터가 우리 생활의 일부로 자리 잡아 가고 있는

현 정보화 사회에서 대부분의 정보들은 인간의 수고를 요하는 타이프라이터나 워드프로세서에 의존하여 데이터로 입력되어져 왔다. 그러나 사회가 점점 다양화되고 전문화 되어감에 따라 그에 따른 정보의 양 또한 급속히 증가하여 기존의 입력방법을 사용하여서는 급증하는 데이터를 신속히 처리할 수 없는 한계

\* 大宥工業專門大學 電氣科  
\*\* 서울시스템(株)  
\*\*\* 光云大學校 컴퓨터工學科  
論文番號 : 93-129

성을 보이고 있다. 따라서 그 한계성을 극복하고 사람과 컴퓨터간의 맨 머신 인터페이스(man machine interface)에 있어서 좀 더 신속한 정보처리를 위한 연구가 활발히 진행 중에 있다.

이러한 시대적 조류에 부응하여 영상 입력 장치를 통한 문서 정보처리 시스템 개발에 많은 관심이 고조되고 있으며, 그 중에서도 패턴인식 분야의 한 부류인 문자인식 분야의 연구는 이미 여러 나라에서 그들의 고유문자에 대해 진행중에 있다. 우리나라의 경우도 한글의 인식에 관한 연구<sup>[1,2]</sup>가 활발히 진행중에 있다. 그러나 한자문화권에 속하는 우리나라와 같이 실제 생활에서 접하는 거의 모든 문서에서 한자를 사용하고 있는 현실을 감안한다면 한자인식에 관한 연구는 매우 미비한 실정에 있다. 같은 한자문화권에 속하는 일본에서는 이 분야에 많은 연구가 있었는데, 특히 문자의 영역을 수치로 표현하기 위하여 여러 가지 벡터의 형태로 표현한 뒤 유사도를 비교하여 인식하는 방법인 결정론적 방법이 많이 연구되어 높은 인식을 얻고 있다. 그외의 인식 방법으로는 문자 패턴에서 문자의 특징점과 스트로크(stroke) 등을 프리미티브(primitive)로 변환하여 여러가지의 문법에 의해 표현한 뒤 이를 비교하여 인식하는 방법인 구분론적 방법과 최근에 새로이 대두되어 연구가 활발히 진행되고 있는 신경망(neural network) 모델을 이용한 방법이 있다. 결정론적 방법으로는 입력 패턴의 주변분포<sup>[3]</sup>를 특징으로 이용하는 방법과 4 방향의 벨트 패턴<sup>[4]</sup>(belt pattern)을 특징으로 이용하는 방법 등이 있다. 주변분포는 디지털 도형의 세로 또는 가로축이 평행한 라인의 흑화소 수를 세는 것만으로 쉽게 구해진 주변분포를 바로 벡터로 표현하거나 또는 퓨리에 변환에 의한 스펙트럼을 벡터로 표현하여 이용한다. 벨트 패턴은 입력 패턴을 블러화 시킨 후 각 방향에서 처음 만나는 흑화소까지의 거리를 벡터로 표현하여 이용하는 방법이다. 이러한 결정론적 방법은 특징값을 추출하기는 용이하나 유사한 패턴이 많은 경우에는 추출된 특징이 모든 경우를 분석하고 표현하기 어려우며 너무나 계산중심적인 면만을 강조한다는 문제점이 있다. 구분론적 방법<sup>[5]</sup>은 문자 패턴에서의 굴곡점과 스트로크의 시작점 및 끝점을 기준으로 하여 프리미티브로 변환하고 표준 패턴과 매칭(matching)시키는 방법으로서, 이 방법은 입력 문자의 글자체에 따라 추출되는 프리미티브가 달라질 수 있는 문제점이 있으며, 특히 필기체 인식의 경우에는 필자에 따라 틀려질 수 있고 또한 같은 필자이더라도

매번 틀릴 수 있으므로 필기체 문자의 인식에는 적합하지 않다.

신경망 모델을 이용한 방법<sup>[6]</sup>은 기존의 인식 방법과는 달리 입력과 출력이 서로 부합(mapping)되도록 노드들간의 연결강도를 조정하는 망을 구성하여 신경망이 스스로 문자의 특징을 찾아내는 방법이다. 이 방법의 경우에는 입력 패턴이 많을 경우 너무 긴 학습 시간을 요구한다는 점과 출력 패턴이 100개 이상일 경우에는 신경망이 포화상태를 보이는 한계성이 있다.

따라서 본 연구에서는 한자패턴과 같이 자종이 광대하고 대상 자형의 구조가 복잡하며 유사 문자가 많은 경우에 대응하기 위하여 결정론적 방법과 신경망을 이용한 방법을 병행하여 한자인식 시스템을 구현하였다. 먼저 한자가 획으로 구성된 문자이고 각각의 획이 방향성을 가지는 특성이 있기 때문에 이러한 특징을 벡터로 표현하였으며<sup>[7]</sup>, 추출된 방향벡터를 한자의 구조적 특성에 의하여 구성된 계층적인 신경망의 입력으로 주어 비교 대상 패턴의 수를 줄이는 분류단계를 행하였고, 인식단계에서는 신경망에 의해 분류된 그룹을 대상으로 방향벡터의 유사도를 비교하여 인식을 하였다. 분류 단계에서의 신경망의 구성은 형식 분류 신경망과 부수 인식 신경망의 두 단계로 나누어지며, 형식 분류 신경망의 입력으로는 입력 패턴에서 추출된 방향벡터 모두를 주어 14 형식 중 하나의 유형으로 1차 분류하였으며<sup>[8]</sup>, 2차 분류로는 각 유형에 따른 부분패턴(부수, radical)의 존재가능 영역에서 추출된 방향벡터만을 부수 인식 신경망의 입력으로하여 부수를 인식함으로써 공통 부수를 갖는 소그룹으로 2차 분류하였다. 인식 단계에서는 2차 분류에서 결정된 소그룹내의 표준패턴들과 부수 인식시 사용된 방향벡터를 제외한 나머지 방향벡터에 대하여 그 유사도를 비교하여 인식을 행하였다.

신경망의 학습 대상으로는 1777자의 문교부 지정 교육용 명조체 한자를 사용하였으며, 실험에 사용한 신경망의 모델은 1개의 은닉층(hidden layer)을 갖는 3층 퍼셉트론(perceptron) 구조로 구성하였고, 개선된 오류 역전파(back-propagation) 학습 방법을 사용하여 각 신경망들을 학습시켰다.

본 논문의 구성은 모두 7장으로 구성되어 있으며, 제2장에서는 한자의 구성 및 형식에 대하여 기술하였고, 제3장에서는 신경망 모델에 대한 개요를, 제4장에서는 신경망의 입력으로 사용한 방향벡터 추출에 대하여, 제5장에서는 문자 인식 시스템에 대하여, 제

6장에서는 실험 결과 및 고찰을 나타내었으며, 제7장에서는 이를 검토하여 얻은 결론을 기술하였다.

## II. 한자의 구조 및 형식

대상 문자인 한자패턴을 인식하기 위한 인식 시스템을 설계하기 위해서는 각 문자의 구조적 특성을 분석하고 그 특징에 따라 인식 시스템을 설계해야 한다. 따라서 본 장에서는 한자의 기본적 구조를 분석하고 이를 이용한 형식 분류에 대해서 알아본다.

### 2.1 한자의 구조분석

대상 문자인 한자패턴을 인식하기 위한 인식 시스템을 설계하기 위해서는 한자의 구조와 구성에 관한 분석이 선행되어야만 한다.

본 연구에서 신경망의 학습 데이터로 삼은 한자는 모필 서체의 이미지를 갖는 명조체인데, 오늘날 한자의 명조체는 모필 서체의 이미지 보다는 금속활자의 발달에 기인한 기계적 명패함을 가진 서체이다. 한자는 한글처럼 조합 문자인데 대개 하나이상의 세그먼트(segment)로 구성된 것이 그 특징이다. 또한 한자는 한글처럼 정방형(사각형)의 형태로 이루어져 있음을 알 수 있다.

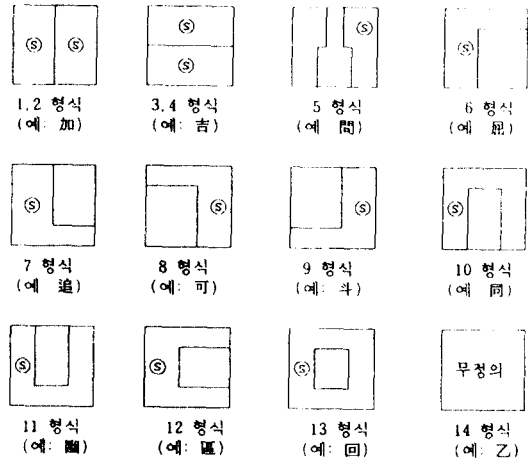
한자의 구조를 살펴보면, “台”와 같은 상하구조(north-south structure), “化”와 같은 좌우구조(east-west structure), “困”과 같은 내외구조(border-interior), 그리고 이 세 구조의 복합구조등으로 구성되어 있는 경우가 아주 많다. 즉, 한자는 상형문자에서 시작되었고 하지만 문명의 발달과 함께 문자에 의해 표현되는 내용이 복잡해지면서 문자의 형태 이미 사용된 형이 조합되기도 하고 원래 하나의 형이던 것이 나누어지는 등 복잡한 형으로 발전 하였다.

이렇듯 한자는 도형적 계층성을 갖는다는 것을 알 수가 있다. 한자는 자종이 광대하고 구조가 복잡하며, 유사문자가 많이 존재하기 때문에 인식은 물론이고 분류하는 데에도 많은 어려움이 따른다. 그래서 이러한 문제점을 극복하기 위해, 한자의 구성 및 구조상의 특징들을 살펴본 결과 한자패턴이 부분패턴을 서로 공통으로 가진 점에 착안하여, 이들이 문자 상에서 위치하는 영역을 중심으로 분류를 하는 방법을 사용하였다.

### 2.2 한자의 형식

본 연구에서는 한자의 도형적 계층성에 의거해 그

림1과 같이 14가지 형식으로 분할을 시도했으며 그 결과를 표1에 나타냈다. 분할방법은 동 표에서처럼 도형적으로 2분할하며 공통 부분패턴을 소유하는 한자의 집합을 동일 그룹으로 분류한다. 단, 『利』, 『推』 등과 같이 2분할된 부분의 어느 것도 공통 부분패턴인 경우에는 동일 그룹내의 한자의 갯수가 많은 쪽으로 분류한다.



Ⓢ : 부분패턴 존재 가능 영역

그림 1. 한자의 14가지 형식

Fig 1. 14 type of Hanja.

표 1. 교육용 한자의 형식별 분류(총 1777자)

Table 1. Classification of educational hanja.

형식	1	2	3	4	5	6	7
갯수	722	165	246	279	11	72	53
형식	8	9	10	11	12	13	14
갯수	13	1	11	2	5	13	184

그림1과 같이 한자를 도형적으로 분류시킨 이유는 한글에서 그 구조적 특성에 의해 크게 6가지 형식으로 구분하여 인식부에 대응을 피했을때 그 유효성을 증명하기가 더 쉬웠다는 점 때문이다.

그러나 한자는 한글에서처럼 그렇게 단순한 구조만을 가지는 것이 아니고 복잡한 구조를 가지는 경우가 많으므로 한글과는 다른 각도에서의 형식 분류가 필요하다. 그러므로 본 연구에서는 대상 한자 패턴이

인쇄체라는 점에 착안하여 대상문자의 구조적 위치를 실험에 의해 정량적으로 분석하여, 신경망에 입력함으로써 대상문자의 형식을 정하고, 다시 분류된 형식으로부터 부분패턴을 인식하는 방법을 사용하여 한자 구조의 복잡성을 해결 할 수가 있을 것이다.

본 연구에 적용할 시스템의 전체적인 흐름도는 그림2에 보였다.

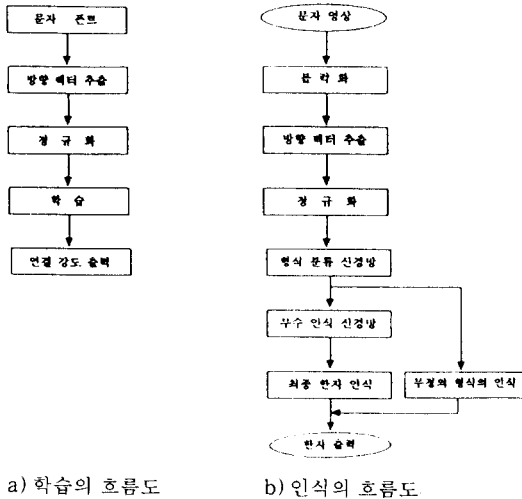


그림 2. 전체 흐름도  
Fig 2. General flow chart.

### III. 신경망 모델의 개요

신경망은 간단한 계산능력을 가진 처리기(노드)들이 서로 복잡하게 연결되어 구성된 컴퓨터 시스템으로서 외부에서 주어진 입력에 대하여 동적인 상태 반응을 함으로써 정보를 처리하는 시스템이다<sup>1)</sup>. 이러한 관점에서 볼 때 신경망 모델은 신경망의 구조, 활성화 함수, 신경망의 학습, 학습 알고리즘의 네 가지로 특성지워진다.

#### 3.1 신경망 모델의 구조

신경망에서 중요한 구성요소는 노드라 불리는 처리기와 노드들 상호간의 연결이다. 노드간의 연결은 보통 방향성 연결선이고 고유의 연결 강도를 가지고 있다. 연결 강도  $W_{ij}$ 는 노드  $i$ 가 노드  $j$ 에 미치는 영향력의 크기를 표시한다. 그러므로, 양의 연결 강

도는 흥분시키는 작용을 하고 음의 가중치는 억제하는 작용을 한다. 연결 강도에 의해 신경망의 동작이 결정되므로 이러한 연결 강도의 결정이 신경망에서 매우 중요한 역할을 한다. 노드는 네트워크상의 역할에 따라 입력 노드, 중간 노드(은닉 노드), 출력노드의 세 부류로 나누어진다. 입력 노드는 다른 노드로부터 들어오는 연결선이 없이 회로망의 외부로부터 입력을 받아들여 활성화되는데, 대부분의 경우 외부의 입력값을 그대로 출력한다. 출력 노드는 회로망 전체의 출력으로 사용되며, 입력 노드나 출력 노드에 속하지 않는 노드를 노드라 부르며, 이는 비선형 분리 함수(non-linearly separable function)와 같은 복잡한 기능을 수행하는 회로망에 필요하다.

신경회로망을 회로망의 구조에 따라 나누어보면 방향성 사이클(directed cycle)이 존재하지 않는 전방향 회로망(feedforward network)과 방향성 사이클이 포함되어 있는 양방향 회로망(recurrent network)이 있다. 신경망 모델의 간략한 구조 그림3에 보였다.

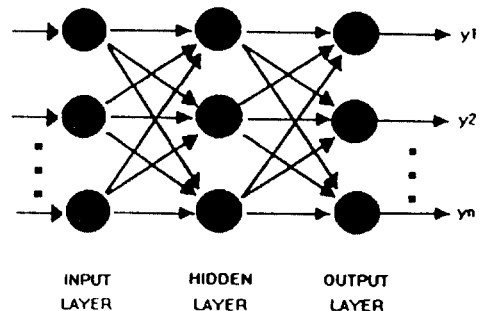


그림 3. 신경망의 구조  
Fig 3. Architecture of Neural Network.

#### 3.2 활성화 함수

신경망에 존재하는 노드는 뇌에서의 신경세포에 해당하며 활성화 값(activation value)이라 불리는 출력값을 계산하는 작용을 한다. 노드는 그림4와 같은 구조로 되어 있으며, 활성화 값은 그 노드에 직접 연결되어 있는 다른 노드들의 출력값과 해당 연결선의 연결 강도를 이용하여 계산한다. 전형적인 신경망 모델에서는 다른 노드들로부터의 입력에 연결선의 연결 강도를 합한 후, 이 결과가 임계치를 초과하는

가를 계산하여, 초과하면 그 값을 그 노드의 출력값으로 한다. 노드의 출력값을 계산하기 위해 사용되는 활성화 함수는 하나의 신경망에서도 노드마다 다른 활성화 함수를 사용할 수도 있다.

본 연구에서는 활성화 함수로 오류 역전파 신경망 모델에서 많이 사용되는 시그모이드 활성화 함수를 사용하였다. 그림5에 시그모이드 함수의 특성 곡선을 나타내었다.

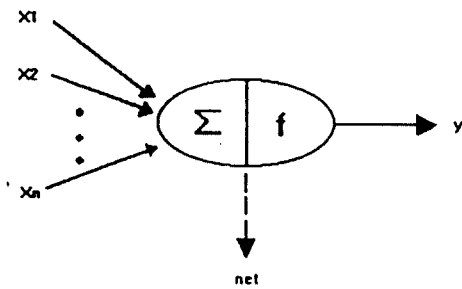
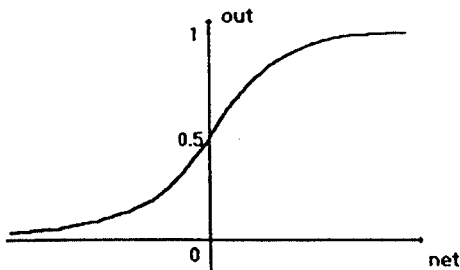


그림 4. 노드의 구조  
Fig 4. Structure of Node.



$$\text{out} = f(\text{net}) = 1/(1+\exp(-\text{net}))$$

그림 5. 시그모이드 함수의 특성 곡선  
Fig 5. Characteristic curve of Sigmoid function.

### 3.3 신경망의 학습

신경망은 데이터를 보관하기 위한 별도의 기억장치를 가지고 있지 않으며, 신경망에서 지식은 노드들이 어떻게 상호 연결되어 있고 각 연결선의 연결 강도가 어떤가에 따라 결정된다. 즉 지식은 특정 장소의 내용이 아니고 신경망의 구조와 연결성에 의해 반영된다. 그러므로 신경망에서 학습이란 주어진 신경망이 원하는 동작을 수행할 수 있는 연결 강도를 찾

아내는 작업을 말한다. 이러한 학습은 입력값과 원하는 출력값의 예를 가지고 연결 강도를 조정해 나가므로써 수행된다. 본 연구에서는 오류 역전파 학습 알고리즘을 이용하여 신경망을 학습시켰다.

### 3.4 오류 역전파 알고리즘

오류 역전파 알고리즘은 지도 알고리즘으로서 이상적인 출력과 실제출력 사이의 평균제곱 오차에 해당하는 비용함수의 값을 최소화 하기위하여 경사 추적 방법을 사용한다<sup>10)</sup>. 이때 노드 사이의 연결 강도를 임의의 작은 값으로 초기화 한 후 학습 데이터를 입력과 출력층에 반복적으로 제공함으로써 학습시킨다. 입력 데이터에 의해서 계산되어 출력된 출력층에서의 값과 기대하고 있던 값과의 차이, 즉 오차값을 하위층으로 전파시키는 방법으로 하위층과의 연결 강도를 재조정해 나가면서 신경망 전체의 연결강도가 수렴하여 총 오차가 매우 작아질 때까지 반복적으로 학습시킨다.

이것은 결국 식(3.2)와 같은 신경망 전체 오차 E를 최소화 하도록 하는 연결 강도를 구하기 위해 식(3.4)와 같은 반복식을 적용하는 것이다.

$$O_{pj} = \sum_i W_{ij} \cdot I_i \quad (3.1)$$

$$E = \sum E_p = \sum \sum (t_{pj} - O_{pj})^2 \quad (3.2)$$

$$\Delta W_{ij}(n) = \eta \cdot (t_{pj} - O_{pj}) \cdot I_i \quad (3.3)$$

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(n) \quad (3.4)$$

여기서  $t_{pj}$ 는 패턴 p의 노드 j의 목표 출력이며,  $o_{pj}$ 는 패턴 p의 노드 j의 실제 출력이며,  $W_{ij}$ 는 노드 i와 노드 j의 연결 강도이고,  $\eta$ 는 학습률(Learning Rate)이며,  $I_i$ 는 노드 i의 입력 값을 나타낸다.

이때 올바른 출력값이 외부에서 주어질 수 있는 출력 노드와는 달리 중간 노드는 그 노드의 올바른 출력값을 알 수 없는 문제가 발생하는데 오류 역전파 알고리즘에서는 선행노드로 오차를 역전파하여 선행노드가 출력 노드의 오차를 책임 분담하는 방법을 사용한다.

## IV. 전처리 및 특징 매트릭스 구성

신경망에 문자를 학습시키고 미지의 문자를 인식하기 위해서는 먼저 문서에 대한 블러화를 행하고 이로부터 각각의 문자를 추출하여 특징 매트릭스를 구

성하여야 한다<sup>[11,12]</sup>. 본 장에서는 문서로부터 문자를 추출하고, 각 문자로부터 특징매트릭스를 구성하는 과정에 대해 설명한다.

4.1 블럭화

문서는 보통 이미지 영역, 문자영역 그리고 그래픽 영역등의 요소들로 구성되어 있다. 본 연구에서는 이러한 영역 중에서 문자부분에 대하여, 각각의 문자 위치를 알기 위하여 문자영역에 대해 블럭화를 행하였다. 블럭화 알고리즘은 오인권의 「Down-up 알고리즘」<sup>[13]</sup>을 개선하여 사용하였다. 그림6은 블럭화된 데이터의 예이다.

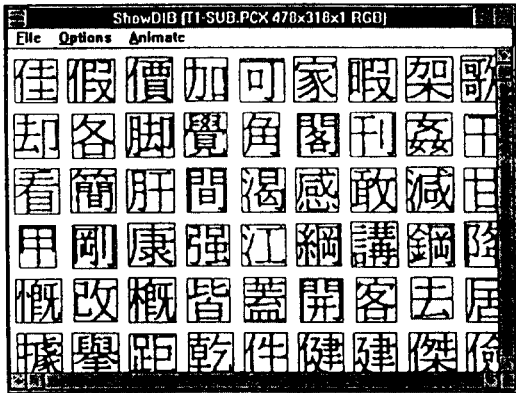


그림 6. 블럭화된 데이터의 예  
Fig 6. An example of a blocked data.

4.2 방향벡터 추출

한자를 구성하는 스트로크들을 살펴보면 대부분의 스트로크들이 수평, 수직, 사선, 역사선 방향의 획 성분은 수평, 수직방향의 획 성분보다 방향벡터값을 찾기 어렵기 때문에 본 연구에서는 사선, 역사선방향에 대해서만 다른 조건을 주어 2차추출을 하였다. 1차추출할 방향벡터 성분은 그림7의 마스크를 이용하여 4방향으로 연속된 흑화소 성분을 조사하여 구하게 되고, 추출된 방향벡터 성분이 수직 또는 수평방향일 경우에는 다음의 식을 만족하는 성분을 구한다.

$$RL > 2 \times W \tag{4.1}$$

$$RL > \sqrt{2} \times W \tag{4.2}$$

$$RL > W \tag{4.3}$$

여기서 RL(run-length)은 추출된 흑화소의 연속길이이고, W(width)는 입력 패턴의 평균 두께를 나타낸다.

식(4.1)과 같이 역치값으로 입력패턴의 평균두께의 두배를 주어 그 보다 적으면 무시하였고, 사선방향이나 역사선방향일 경우에는 식(4.2)와 같이 역치값을 좀더 낮게 잡아 비교하였다. 1차추출된 4방향벡터는 원래의 입력패턴에서 제외되며, 사선방향과 역사선방향의 방향성을 강조하기 위해 1차추출 조건으로는 구하기 어려운 점성분이나 빗침성분과 같은 짝

③(x-1, y-1)	②(x, y-1)	①(x+1, y-1)
④(x-1, y)	⑥(x, y)	⑧(x+1, y)
⑤(x-1, y+1)	⑦(x, y+1)	⑨(x+1, y+1)

- 주 목 점 : ⑥
- 사 선 방 향 : ① ⑤ 방향
- 수 직 방 향 : ④ ⑧ 방향
- 역사선 방 향 : ③ ⑦ 방향
- 수 평 방 향 : ② ⑨ 방향

그림 7. 방향벡터 추출 마스크  
Fig 7. Mask for extracting directional vector.

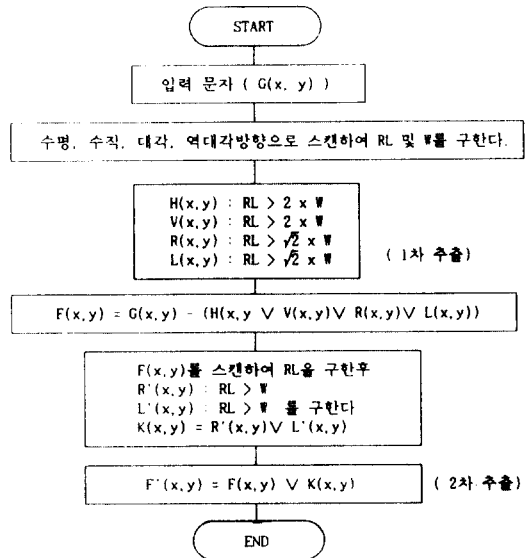
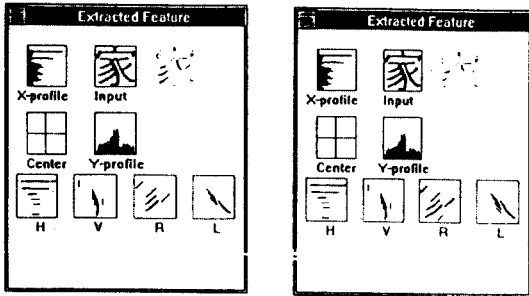


그림 8. 방향벡터 추출 알고리즘  
Fig 8. Algorithm for extracting each direction vector.

은 사선성분에 대하여 식(4.3)과 같이 역치값을 입력 문자패턴의 평균 두께로 주어 2차추출을 행하였다. 이를 이미 추출된 1차방향벡터와 합하여 입력된 문자 패턴에 대해 추출된 전체방향벡터로 하였다. 방향벡터를 구하는 알고리즘을 그림8에 흐름도로 보였으며, 그림9에는 알고리즘을 적용하여 추출된 방향벡터를 보였다.



(a) 1차 추출의 예 (b) 2차추출의 예

그림 9 방향벡터의 추출  
Fig 9. Extracting of directional vector.

4.3 특징 매트릭스 구성

추출된 방향벡터를 신경망의 입력으로 사용하기 위해서는 신경망을 구성했을 경우에 요구되어지는 메모리와 학습에 소요되는 시간을 고려하여야 한다.

본 연구에서는 신경망에 요구되어지는 메모리와 계산량을 줄이고 입력문자 패턴의 부분별 복잡도에 따른 특징을 추출하기 위하여 입력된 문자 패턴의 무게 중심에 따라 가로, 세로방향에 대하여 각각 7개의 분류점을 구하므로써 64개의 메쉬 블럭을 구성하여 추출된 각 방향벡터에 씌운 다음 각 메쉬내의 흑화소 라인의 길이를 구하여 각 방향에 따른 4개의 8×8 특징 매트릭스를 구성하였다. 그림10은 7개의 분류점을 구하여 64개의 메쉬를 분할하는 알고리즘을 흐름도로 보였다. 이 알고리즘을 적용하여 메쉬로 분할된 방향벡터의 예는 그림 11이고, 각 메쉬내의 방향벡터를 모두 합한 값으로 표현하여, 추출한 특징 매트릭스의 예를 그림 12에 보였다.

4.4 흐림화(blurring)처리

입력된 문자패턴의 잡음에 따른 무게 중심의 변화로 인해 메쉬의 구성이 달라지므로써 추출되는 방향

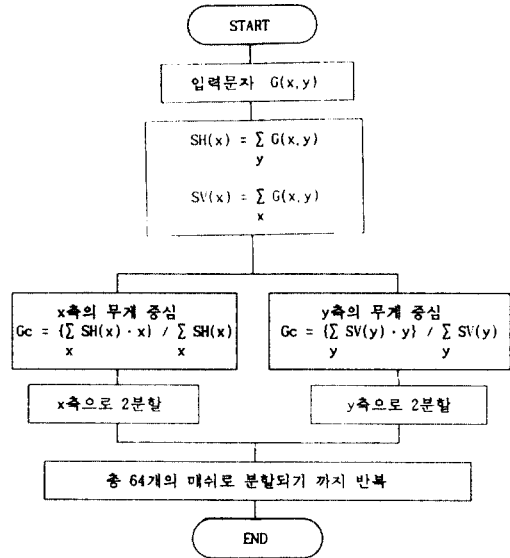


그림 10. 메쉬 추출 알고리즘  
Fig 10. Algorithm for extracting mesh.

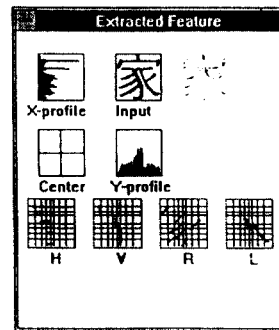


그림 11. 메쉬로 분할된 방향벡터  
Fig 11. Directional vector divided by mesh.

Vector H								Vector V							
3	7	5	5	4	6	7	7	0	0	0	0	0	0	0	0
4	7	5	5	4	6	7	7	0	0	0	0	0	0	0	0
2	7	5	5	4	6	6	0	0	0	0	1	0	0	0	0
2	9	10	10	5	6	6	0	0	0	0	13	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	9	5	2	0	0
0	0	0	0	0	0	0	0	0	0	0	6	7	6	0	0
0	0	0	0	0	0	0	0	0	0	0	1	7	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	7	6	0	0

(a)수평 방향 (b)수직 방향

Vector R							
0	1	0	0	0	0	0	0
9	3	0	0	0	0	0	0
6	0	0	3	3	0	0	0
0	13	10	14	4	0	7	5
0	2	7	9	5	5	10	0
1	118	108	6	10	0	0	0
1	9	107	0	0	0	0	0
3	102	0	0	0	0	0	0

Vector L							
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	2	0	0	0	0	0
0	0	6	103	4	0	0	0
0	0	0	7	9	9	0	0
0	0	0	1	5	117	0	0
0	0	0	0	0	1	125	0
0	0	0	0	0	0	3	7

(c) 사선 방향 (d) 역사선 방향

그림 12. 추출된 특징 매트릭스  
Fig 12. Extracted feature matrix.

벡터의 값과 학습시킨 표준패턴 사이의 위치 변화에 대응하기 위하여, 추출된 특징 매트릭스 값을 그림 13의 마스크를 이용하여 보정해주어 잡음에 대한 적응력을 가질 수 있도록 처리하였다<sup>[14]</sup>. 그림 14에 흐림화 처리를 한 특징 매트릭스 값을 보였다. 흐림화 마스크에 주어진 값은 흐림화 처리에서 가장 많이 사용되어지는 값이다.

1	2	1
2	Ⓢ	2
1	2	1

그림 13. 흐림화 처리에 사용된 마스크  
Fig 13. Mask for blurring process.

#### 4.5 정규화

문서 중에는 다양한 문자의 크기를 갖는 문자들이 존재하므로 이에 대처하기 위해서 정규화를 거쳐야 하는데, 지금까지 발표된 한자인식에 관련된 연구에서는 문자의 다중크기에 대응하기 위해서 입력된 문자패턴에 대한 전처리로서 정규화를 시킨 뒤 특징을 추출하여 인식에 사용하여 왔다. 본 연구에서는 기존의 복잡한 정규화에 의해서 발생할 수 있는 문자의 변형을 막고 보다 쉽게 입력 문자패턴의 크기 변화에 대응하기 위하여 전처리로서 정규화를 행하지 않고, 특징 매트릭스를 구한 뒤 흐림화 처리를 한 후 입력 문자 패턴의 가로, 세로폭을 구하여 식(4.4)~(4.7)과 같은 간단한 처리를 해주므로서 정규화를 행하여 전처리에 소요되는 시간을 줄였다<sup>[9]</sup>.

그림 15는 정규화 처리를 행한 특징 매트릭스의 예이다.

$$H(m,n) = \{KH(m,n) \times kc\} / \Delta X \quad (4.4)$$

$$V(m,n) = \{KV(m,n) \times kc\} / \Delta Y \quad (4.5)$$

$$R(m,n) = \{KP(m,n) \times kc\} / \{(\Delta X + \Delta Y) / 2\} \quad (4.6)$$

$$L(m,n) = \{KL(m,n) \times kc\} / \{(\Delta X + \Delta Y) / 2\} \quad (4.7)$$

여기서 KH(m,n), KV(m,n), KL(m,n), KP(m,n)은 블러링 처리 후 특징 매트릭스값이며, H(m,n), V(m,n), R(m,n), L(m,n)은 정규화 처리 후 특징 매트릭스값이며, kc는 정수값을 만들기 위한 상수값이고, Δ X, Δ Y는 입력 문자패턴의 가로, 세로폭이다.

Vector H									
53	95	86	77	73	93	10	91		
70	11	10	96	92	11	12	97		
58	12	12	11	99	11	10	59		
45	11	14	4	10	98	79	18		
13	34	69	99	84	47	22	6		
0	2	17	56	58	30	5	0		
0	1	11	27	28	14	3	0		
0	2	18	50	44	16	2	0		

Vector V									
28	10	0	0	0	0	0	0	0	0
56	16	1	2	1	0	0	0	0	0
28	10	11	30	19	4	0	0	0	0
4	2	24	97	71	26	11	3		
0	0	25	91	10	97	58	12		
0	0	18	86	13	16	92	16		
0	0	8	48	12	12	52	8		
0	0	1	23	90	84	21	1		

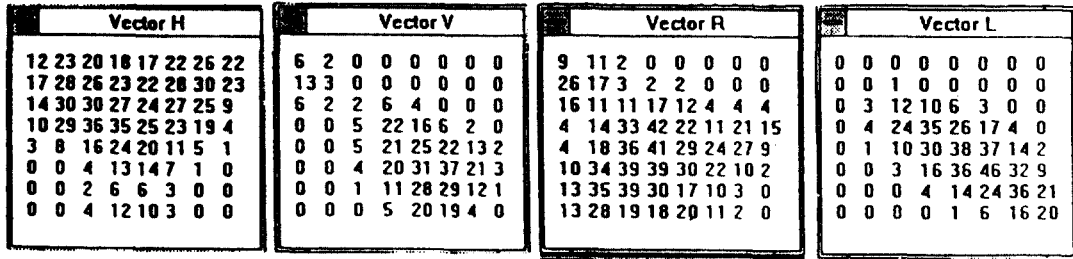
Vector R									
41	47	11	0	0	0	0	0	0	0
11	74	16	9	9	3	0	0	0	0
70	47	48	72	52	17	19	17		
18	60	14	17	93	50	91	64		
20	77	15	17	12	10	11	38		
43	14	16	16	12	193	46	12		
55	15	116	13	72	45	14	1		
55	11	90	76	88	50	9	0		

Vector L									
0	0	0	0	0	0	0	0	0	0
0	3	6	3	0	0	0	0	0	0
0	14	52	44	28	14	4	0		
0	19	10	4	1	73	20	1		
0	8	45	12	16	15	62	10		
0	0	13	68	15	19	13	3		
0	0	3	20	59	10	11	5		
0	0	0	2	7	26	70	84		

(a) 수평 방향 (b) 수직 방향 (c) 사선 방향 (d) 역사선 방향

그림 14. 흐림화 처리후의 특징 매트릭스  
Fig 14. Feature matrix after blurring process.





(a) 수평 방향 (b) 수직 방향 (c) 사선 방향 (d) 역사선 방향

그림 15. 정규화된 특징 매트릭스의 예  
Fig 15. Example of normalized feature matrix.

### V. 문자 인식 시스템

패턴인식에 널리 쓰이고 있는 오류 역전과 신경망은 구별해야 될 패턴의 수가 증가할수록 인식이 저하되며, 더 많은 중간 노드를 필요로 하게 되므로 학습에 필요한 시간도 더욱 많아지게 된다. 본 연구는 조 성배가 제안한 계층적 신경망을 기본으로 하여 두 단계의 신경망을 구성하여 패턴에 대한 분류를 행하였고, 마지막으로 최종인식 단계를 두어 인식을 행하였다.

#### 5.1 시스템의 구성

본 연구에 사용된 시스템은 신경망을 두 단계로 나누어, 문자의 입력이 주어지면 먼저 형식분류 신경망이 입력된 문자를 14가지 유형 중의 하나로 대분류하고, 이 형식분류 신경망이 입력된 문자의 부수를 인식하여 부수별로 소분류한 뒤, 마지막으로 소분류된 그룹내의 문자들과 유사도를 비교하여 최종인식하는데 이를 흐름도로 그림16에 나타내었다.

형식분류 신경망에서는 한 출력노드에 같은 형식의 문자들만을 대응시키므로서 분류의 다음 단계인 부수인식 신경망에 같은 형식의 문자들만을 입력의 형태로 제한할 수 있고 이를 다시 같은 부수를 갖는 문자들만 하나의 출력노드로 대응시키므로서 두차례의 분류단계를 두어 신경망을 학습시킴으로써 학습속도를 개선하였고, 최종 인식부에서는 부수인식에 사용된 나머지 부분에 대하여만 유사도록 계산하여 인식하므로서 인식율을 높일 수 있다.

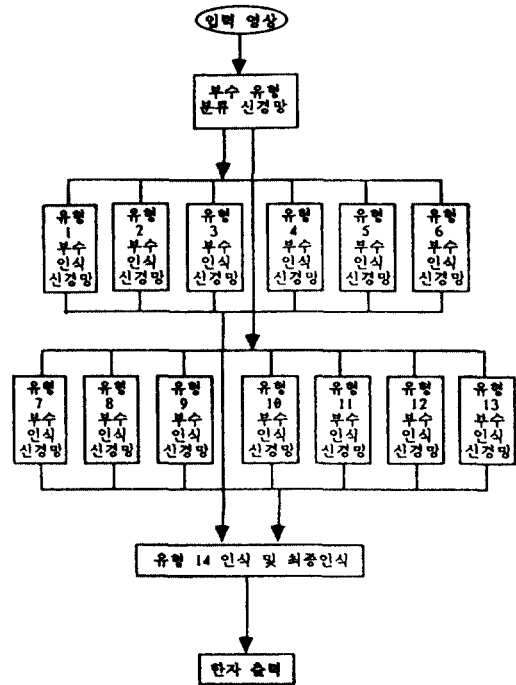


그림 16. 한자 인식 시스템  
Fig 16. System for recognize of Hanja.

#### 5.2 형식 분류신경망

형식 분류신경망의 입력으로는 제4장에서 추출한 특징 매트릭스를 모두 입력으로 사용하였다. 64개의 메쉬마다 4방향의 방향벡터가 있으므로 모두 256개의 방향벡터가 형식 분류신경망의 입력노드로 들어간다. 형식 분류는 부수의 존재 가능 위치에 따라 14

형식으로 나누었으므로 출력노드는 14개가 존재한다. 여기서 25개의 중간노드를 두어서 형식 분류에 사용하였다. 이때 형식 분류신경망은 모두  $256 \times 25 + 25 + 25 \times 6 = 6750$ 개의 연결선을 갖는다.

5.3 부수 인식신경망

부수 인식신경망은 무형식인 14형식을 제외한 13개의 형식별로 각각 따로 존재하며 같은 형식을 갖는 문자패턴들은 유사한 위치에 부수패턴들이 존재하므로 부수패턴들이 존재할 수 있는 영역을 전체문자패턴에서 분리하여 해당 부수 인식신경망의 입력으로 사용하였다. 그렇게 하므로서 신경망의 규모를 축소시키고 학습 및 인식시에 소요되는 계산량과 시간을 단축시킬 수 있다. 출력노드는 해당형식이 갖는 부수의 갯수 만큼을 가진다. 그림17에 각 형식별 부수영역의 분리 위치를 보였다.

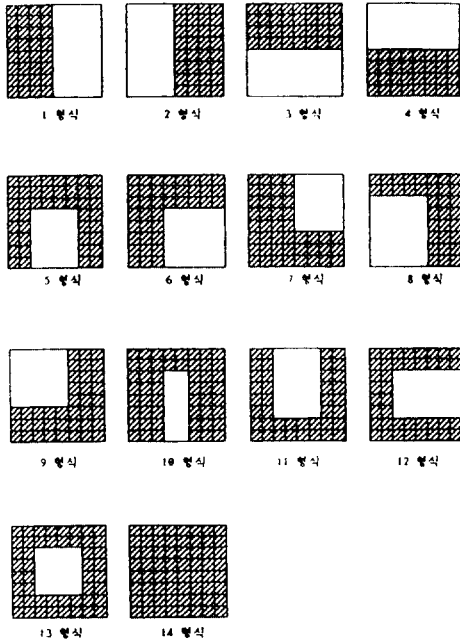


그림 17. 형식별 부수 분리 위치  
Fig 17. Position of separated BUSU for each types.

5.4 무정의 형식의 인식 및 최종인식

부수가 인식된 후의 최종인식과 무정의 형식에 대한 인식은 제4장에서 구한 특징 매트릭스를 이용하여

행하게 된다. 부수인식을 하여 부수에 따른 소그룹으로 분류된 문자패턴은 부수인식시 사용된 부분을 제외한 나머지 부분의 방향벡터값에 대해서만 유사도를 비교하며, 무정의 형식의 경우는 전체 방향벡터값에 대하여 유사도를 비교하여 해당 그룹내에서 가장 작은 차이를 보이는 문자패턴을 취하므로서 인식을 행한다. 입력 문자패턴의 특징 매트릭스  $V(i, j)$ 와 표준패턴의 특징 매트릭스  $M(i, j)$ 에서 유사도는

$$d = \sum_{i=1}^m \sum_{j=1}^n |V(i, j) - M(i, j)| \tag{5.1}$$

또는

$$d = \sum_{i=1}^m \sum_{j=1}^n (V(i, j) - M(i, j))^2 \tag{5.2}$$

여기서  $V = (V_1, V_2, \dots, V_m)$ 는 입력 문자패턴의 매트릭스값이며,  $M = (M_1, M_2, \dots, M_n)$ 은 표준 문자패턴의 특징 매트릭스값이며,  $m, n$ 은 각 형식별 차원수(사용 매트릭스의 x축과 y축의 분할 수)이고,  $d$ 는 거리이다.

단 표준 문자패턴은  $48 \times 48$  크기의 명조체 한자폰트이다. 여기에서 유사도는 거리에 반비례하는 것으로 유사도가 크기 위해서는 거리값이 적어야 한다. 패턴이 완전히 매칭이 되는 경우에는 거리값은 0이 된다. 표2에는 최종 인식에 사용되는 특징매트릭스의 형식별 차원수를 보였다.

표 2. 각 형식별 거리 비교 차원수

Table 2. Dimensional number of distance to compare for each type.

구분	1형식	2형식	3형식	4형식	5형식	6형식	7형식
차원수	128	128	128	128	80	100	80
구분	8형식	9형식	10형식	11형식	12형식	13형식	14형식
차원수	120	100	48	96	96	64	256

5.5 신경망의 크기

신경망의 크기는 문자 인식시스템에 필요한 메모리의 양과 학습 및 인식에 소요되는 계산량 및 시간에 큰영향을 주기 때문에 인식 효율에 영향을 주지않는 한, 신경망의 크기를 될수록 작게 구성할 필요가 있다. 각 신경망은 문자패턴에서 입력시킬 부분과 출

력될 형식에 따른 부수의 수가 정해지므로서 입력노드와 출력노드의 수가 결정된다. 그러므로 중간노드의 수를 조절하므로서 신경망의 크기를 조절할 수 있다.

본 연구에서는 중간노드의 수를 형식분류 신경망의 경우는 25개로 초기 설정하고 부수 인식 신경망은 15개로 설정하여 학습을 시킨 다음, 수렴이 불가능하게 될 때까지 중간노드의 수를 하나씩 줄여가면서 계속하여 학습을 시키는 방법<sup>[15]</sup>으로 중간노드의 수를 조절하였고, 신경망의 크기를 결정하였다. 표3에 형식 분류신경망과 부수인식신경망의 크기를 보였다.

표 3. 신경망의 크기

Table 1. Size of Neural Network.

구분	형식	1형식	2형식	3형식	4형식	5형식	6형식	7형식
입력	256	128	128	128	128	176	156	176
중간	25	15	15	15	15	10	10	10
출력	14	66	32	41	62	2	30	10

구분	8형식	9형식	10형식	11형식	12형식	13형식
입력	136	156	208	160	160	192
중간	10	10	10	10	10	10
출력	18	8	10	3	3	2

5.6 신경망의 학습

각 신경망의 학습을 위하여 오류역전파 알고리즘을 사용하고, 신경망의 학습시간을 단축하기 위하여 기본적인 오류 역전파 학습방법에 바이어스와 관성항 파라메타를 첨가하여 학습을 행하였으며, 「Decending Epsilon」 알고리즘<sup>[16]</sup>을 적용하여 Epsilon을 0.5에서 0.05까지 감소시켜 나가면서 각 신경망을 학습시켰다.

이전의 연결강도 변화량의 경향을 다음 단계에도 반영시키는 역할을 하는 파라메타가 관성항( $\alpha$ )이며, 바이어스(Bias)는 항상 1인 값을 가상입력으로서 노드의 임계치를 조정하는 효과를 내기 때문에 학습과정을 보다 빨리 수렴에 도달하도록 한다. 또한 바이어스의 연결강도는 노드의 연결강도와 마찬가지로 학습이 가능하므로 본 연구에서는 노드의 연결강도와 바이어스의 연결강도를 같은 방법으로 조정하였다.

$$\Delta W_{ij}(n+1) = \eta \times \delta_{pj} \times O_{pi} + \alpha \times \Delta W_{ij}(n) \quad (5.3)$$

여기에서  $\Delta W$ 는 연결강도의 변화량이고,  $\delta$ 는 역방향으로 전파된 오차의 분담값이며, 0는 노드의 출력값이다. 또한 파라메타  $\eta$ 는 학습율(learning rate)로서 이 값이 클수록 일반적으로 학습을 빨리 마칠 수 있으나, 수렴을 못하고 진동하는 경우가 발생하므로 학습율은 진동이 일어나지 않는 범위에서 최고값을 주는 것이 바람직하다.

VI. 실험 및 고찰

6.1 실험 시스템

본 연구에서 사용한 실험 시스템은 Intel-80387 수치연산 보조 프로세서를 장착한 IBM 389-PC 호환 컴퓨터를 사용하여 MS-WINDOW 환경에서 실험하였으며, 학습 데이터는 48×48 크기의 명조체 한자 폰트를 Hewlett Packard의 LaserJet-II P 레이저 프린터(laser printer)를 이용하여 인쇄한 후에 Hewlett Packard의 ScanJet-Plus 이미지 스캐너(image scanner)로 문자를 입력하여 사용하였으며, 인식 데이터는 같은 과정을 다시 반복하여 사용하였다. 또한 실험된 데이터의 출력은 Hewlett Packard의 LaserJet-II P를 이용하였다.

6.2 실험 결과

1777자의 문자를 14형식에 의하여 1차분류한 결과 94.4%의 분류율을 얻었고, 형식이 올바로 분류된 문자를 대상으로 부수인식에 의하여 2차분류한 결과 90.4%의 분류율을 얻었으며, 부수까지 올바로 분류된 문자에 대하여 96%의 인식율을 얻었다. 시스템의 처리속도는 형식분류와 문자 인식을 포함하여 1초당

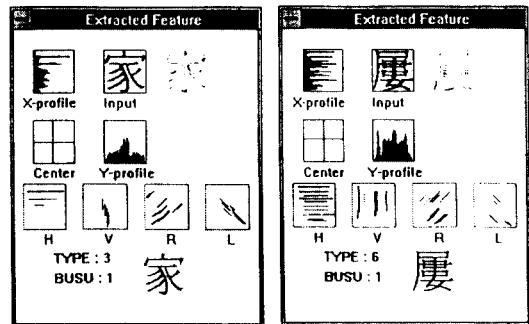


그림 18. 인식된 데이터의 예

Fig 18. An example of a recognized data.



하여 인식을 행하였다. 한글 문자의 획 성분의 방향 벡터를 입력으로 하는 신경망을 이용한 분류단계는 신경망의 특성의 병렬처리에 의해 분류까지의 시간을 기존의 방법들보다 줄일 수 있었으며, 분류율에 있어서는 다소 효율이 저조하였다. 후처리로서 구별하기 어려운 문자에 대하여 소규모의 신경망을 따로 구성하여 주거나 부수 인식신경망의 입력을 좀 더 세분화하여 부가해주므로서 더 좋은 인식율을 얻을 수 있을 것이다. 또한 분류 단계에서의 무게중심에 의한 매쉬의 구성을 효율적으로 이용하면 다양한 글자체에 대해서도 적용할 수 있도록 신경망을 개선할 수 있을 것이다. 앞으로는 다중 크기 및 다중 활자체에 제한이 없는 한자인식에 대한 연구와 더불어 이를 하드웨어화하여 문자인식의 실시간 처리에 적용할 수 있도록 하는 연구도 계속되어야 할 것이다.

### 참 고 문 헌

1. 남궁재찬, "Index-Window 알고리즘에 의한 한글 pattern의 부분 분리와 인식에 관한 연구," 인하대학교 박사학위 논문, 1982.
2. 이주근, 남궁재찬, 김영건, "한글 Pattern에서 Subpattern 분리와 인식에 관한 연구," 대한전자공학회지, Vol.18, No.3, pp.1-8, 1983.
3. Yasuaki Nakano, Kazuo Nakata, "Recognition of Chinese characters using peripheral distributions and their amplitude spectra," 日本電子通信學會 論文誌, Vol.56-D, No.3, pp.47-59, 1973.
4. C. Y. Suen, "Computer recognition of KANJI characters," Proc. Intel. Conf. on text processing with a large character set, Oct. pp. 429-435, 1983.
5. Chia-Wei Liao, Jun S. Huang, "A transformation invariant matching algorithm for handwritten Chinese character recognition," Pattern recognition, Vol.23, No.11, pp.1167-1188, 1990.
6. 이회건, 방승양, "신경회로망을 이용한 오프라인 문자인식 연구 동향," 정보과학회지, Vol.9, No.1, 1991.
7. Yamashita Y., Tegakikanji no Daibunruihou, "Classification of handwritten KNAJI," OKI technical review, No.121, Vol.50, pp.72-82, 1983.
8. 김정환, "인쇄체 한자에서 Radical의 구조적 정보를 이용한 형식분류 및 부분패턴 추출에 관한 연구," 광운대학교 대학원 석사 학위 논문, 1990.
9. 이회건, 방승양, "신경회로망을 이용한 오프라인 문자인식 연구 동향," 정보과학회지 Vol.1, No.1, 1991.
10. 조성배, 김진형, "오류역전과 학습 알고리즘의 속도 개선방법: 분류 및 비교," 한국정보과학회 논문지, Vol.18, No.6, 1991.
11. Yamashita, et al., "Classification of KANJI using the structured line element conformity method," Journal of communication PRL 81, No.93, pp.85-92, 1981.
12. Yamashita, et al., "Classification of KANJI using the structured line element conformity method-Use of periperal distribution" Journal of communication PRL 82, No.12, pp.25-30, 1982.
13. 오인권, "영문이 혼합된 한글 문서에서의 문자 및 특수 문자 추출에 관한 연구," 광운대학교 대학원 석사 학위 논문, 1988.
14. Iijima, "The theory of character recognition using matching," Journal of communications, Vol.56, No.5, pp668, 1973.
15. Yoshio Hirose, Koichi Yamashita, Shimpei Hijiya, "Back-propagation algorithm which varies the numbers of hidden uits," Networks, Vol.4, pp.61-66, 1991.
16. Yeong-ho Yu, Robert F. Simmons, "Decending Epsilon in back-propagation: A technique for better generalization," Artificial Intelligence laboratory, Univ. Texas, Austion, pp.167-172, 1990.

李 成 範(Sung Bum Lee)

正會員

1973년 : 한양대학교 전기공학과 졸업(공학사)  
1981년 : 동국대학 대학원 졸업(공학석사)  
1992년 : 광운대학 대학원 전산기과 박사수료  
1981년 ~ 현재 : 대유공업전문대학 전기과 부교수  
※주관심분야: 패턴인식, 인공지능, 지식공학

吳 鍾 旭(Jong Wook Oh)

正會員

1990년 : 광운대학교 전자계산기공학과 졸업(공학사)  
1992년 : 동대학 대학원 컴퓨터공학과 졸업(공학석사)  
1992년 ~ 현재 : 서울시스템(주)  
※주관심분야: 패턴인식, 컴퓨터 비전

南宮 在 贊(Jae Chan Namkung)

正會員

1970년 : 인하대학교 전기공학과 졸업(공학사)  
1976년 : 동대학 대학원 전자공학과 졸업(공학석사)  
1982년 : 동대학 전자공학과 졸업(공학박사)  
1982년 ~ 1984년 : 일본 Tohoku대학 객원교수  
1979년 ~ 현재 : 광운대학교 컴퓨터공학과 교수  
※주관심분야: 패턴인식, 컴퓨터 비전, 인공지능