

특징점 추출에 의한 한글 문자 인식 및 전처리용 신경 칩의 설계

(Korean Character Recognition by the Extraction of Feature Points and Neural Chip Design for its Preprocessing)

金 琮 烈*, 鄭 鎬 宣**, 李 宇 一**

(Jong Ryoul Kim, Ho Sun Chung, and Wu Il Lee)

要 約

본 논문에서는 특징점을 이용한 한글 문자를 인식 방법에 대해서 연구하였다. 또한 잡음 제거, 선형화, 세선화 및 특징점 추출등의 전처리용 신경 chip을 설계하였다. 자소의 분리는 개선된 index 알고리즘을 사용하였고, 특징점 분류 방법으로 인식하였다. 실험결과 한글 자소의 분리율은 약 97% 이었으며 인식율은 95%이었다. 설계된 전처리용 chip은 SPICE를 이용하여 시뮬레이션하였고 CMOS 이중 금속 2μm 설계 규칙에 따라 레이아웃하였다.

Abstract

This paper describes the method of the Korean character recognition by means of feature points extraction. Also, the preprocessing neural chip for noise elimination, smoothing, thinning and feature point extraction has been designs. The subpatterns were separated by means of advanced index algorithm using mask, and recognized by means of feature points classification. The separation of the Korean character subpatterns was obtained about 97%, and the recognition of the Korean characters was obtained about 95%. The preprocessing neural chip was simulated on SPICE and layouted by double CMOS 2μm design rule.

I. 서 론

한글 문자는 정보의 수집 및 저장의 수단으로써 그 이용도가 날로 증가하고 있다. 따라서 한글 문자로 된 정보를 신속하게 처리하기 위하여 한글 문자 인식 시스템을 개발하여 문서 내용의 입력등 사람에

의존하는 작업을 기계로 대신하려는 연구가 활발히 진행되고 있다.^{[1]-[5]}

기존의 한글문자 인식 방법에는 원형 정합^[1] 구조적 해석 방법,^[2] 신경회로망을 이용한 방법^[4,5]등이 있다. 이러한 인식 방법들은 주로 소프트웨어에 의존하여 실시간적인 하드웨어 구현을 고려하지 않아서 실용화 되기는 어렵다. 따라서 한글 문자의 실시간 인식을 위해서는 인식 과정에서 시간이 많이 소요되는 부분을 하드웨어로 구현해야 할 필요가 있다.

본 논문에서는 이미지 스캐너에서 입력된 한글문자를 그 특징점을 이용한 구조적 해석 방법으로 인식하였다. 이 방법을 쓰면 마스크를 이용하여 특징

*正會員, 韓國電氣通信公社
(Korea Telecommunication Authority)
**正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)
接受日字: 1990年 1月 6日

점을 쉽게 추출할 수 있으며 자소의 인식도 쉬운 장점을 가지고 있다. 마스크는 단층 구조 퍼셉트론의 개념으로 하드웨어로의 구현이 용이⁶⁾하다. 또한 문자 인식 속도의 한계성을 극복하기 위해 한글 문자 인식에서 소프트웨어적으로 많은 시간이 소요되는 전처리 과정을 하드웨어적으로 수행할 수 있는 칩에 수반되는 소프트웨어를 개발하였다. 본 연구에서 설계된 전처리 과정의 회로는 잡음 제거, 선형화, 세션화 및 특징점 추출의 과정을 수행한다. 설계된 모든 회로는 SPICE 시뮬레이션 하였으며, 이중 금속 CMOS 2 μ m 설계규칙으로 KUIC-LED (kyungpook national university intelegent CAD-Layout Editor)¹⁸⁾ 상에서 레이아웃하였다.

II. 소프트웨어 구성

한글 문자를 인식하기 위한 전체적인 소프트웨어 구성은 그림1에서와 같다. 그림1에서 카메라나 이미지 스캐너와 같은 영상 입력 장치를 통하여 입력된 영상 정보는 입력장치의 오류 혹은 문서의 파손등으로 불필요한 잡음을 가지기 쉽다. 이러한 잡음은 인식의 효율을 저하시키므로 제거시킬 필요가 있다. 그래서 문자 인식과정을 수행하기 전에 먼저 전처리 과정에서 이러한 불필요한 잡음을 없애고 원하는 형태의 데이터로 문자 영상을 개선 및 변환시켜야 한다. 이렇게 개선된 문자 영상에 대해 인식과정을 수행한다.

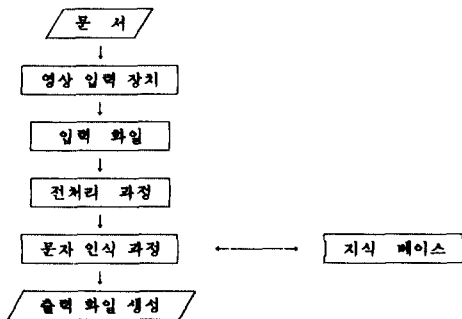


그림 1. 문자 인식 시스템의 구성
Fig. 1. The configuration of korean character recognition system.

III. 전처리 과정

1. 개별문자 분리의 과정

그림2에서는 이미지 스캐너를 이용하여 해상도 250 DPI(dot per inch)로 받은 문자 영상이다. 일반적으로

로 문서는 그림2와 같이 문자열의 집합속에 개별 문자를 가지고 있다. 따라서 문자인식을 위해서는 먼저 문자열에서 부터 개별문자를 분리한 후, 이들 개별문자에 대해 순차적으로 문자를 인식하게 된다. 개별문자를 분리하기 위한 알고리즘은 다음과 같다. 먼저 좌표점(0.0)에서 부터 하나의 수평 라인에 문자를 형성하는 오브젝트 화소가 나타날때까지 추적한다. 오브젝트를 이루는 화소가 처음으로 나타나는 점의 y좌표값이 하나의 문자열이 시작되는 y축 좌표값이다. 이 좌표값을 iy라고 둔다. 그 다음에는 이 y좌표로 부터 y축 좌표값을 증가시키면서 하나의 수평라인 전체에 걸쳐 문자를 이루는 오브젝트 화소가 하나도 없는 최초의 y좌표까지 증가시킨다. 이때의 y좌표가 하나의 문자열이 끝나는 y축 좌표값이다. 이 좌표값을 ly라고 둔다. 이런 알고리즘으로 하나의 문자열이 존재하는 y축의 좌표 (iy, ly)를 찾을 수 있다. 이렇게 하나의 문자열이 분리되면 다음에는 수직으로 문자를 분리하여 문자열에서 개별 문자들을 분리한다. 먼저 문자열이 시작되는 y좌표, 즉 iy에서 하나의 문자열이 끝나는 y좌표, 즉 ly까지 문자열의 제일 왼쪽에서부터 y축 방향으로 추적한다. 이렇게 추적하다가 문자를 형성하는 오브젝트 화소가 처음으로 나타나면 이 점의 x좌표값이 하나의 문자가 시작되는 x축 좌표값이다. 이 값을 ix라 둔다. 다음에는 이 x좌표로 부터 x좌표값을 증가시키면서 iy에서 ly까지의 수직라인 전체에 걸쳐 문자를 이루는 오브젝트 화소가 하나도 없는 최초의 x좌표를 찾

고향 고향 내 고향,
 박꽃 피는 내 고향.
 담 밑에 석류 익는
 아름다운 내 고향.
 고향 고향 내 고향,
 바다 푸른 내 고향.
 석양에 놀을 따라
 물새 우는 내 고향.

그림 2. 스캐너에 의해 입력된 문자영상
Fig. 2. Scanned character image by image scanner.

는다. 이때의 x좌표가 이 문자가 끝나는 x축 좌표 값이다. 이 값을 lx 라 둔다. 이렇게 하여 하나의 개별 문자가 존재하는 영역 (ix, iy) 에서 (ly, ly) 까지를 찾을 수 있다. 이상의 알고리즘을 입력된 문자 영상 전체에 걸쳐 수행시키면 입력된 문자열에서 개별 문자를 분리하여 문자를 인식할 수 있다.

2. 잡음 제거의 과정

영상 입력 장치를 통하여 입력된 영상정보는 입력 장치의 오류나 문서의 파손등으로 불필요한 잡음을 갖기 쉽다. 이러한 잡음은 인식의 효율을 저하시키므로 제거해야 한다. 본 논문에서는 2개 이하로 연결된 화소를 잡음으로 간주하고 이를 제거하였다.

3. 세션화 과정(9)

영상입력 장치로 부터 입력된 문자 영상은 그림 2에서처럼 문자의 선폭이 두꺼운 선폭으로 나타난다. 세션화 과정이란 이렇게 두꺼운 선폭으로 나타나는 문자의 선폭을 하나의 화소폭으로 변화시키는 과정을 말한다. 그림3은 입력된 문자영상에서 개별 문자를 분리하고 세션화한 결과이다.

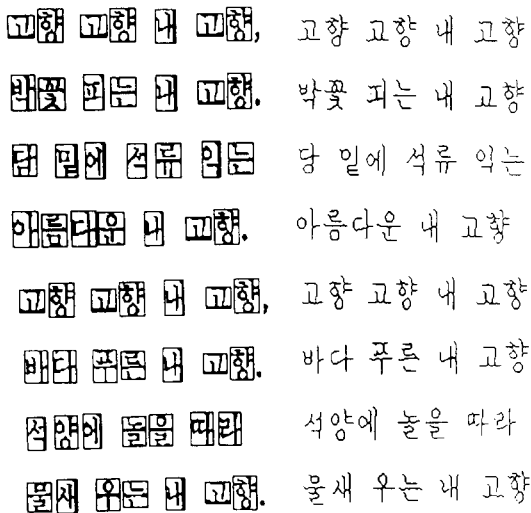


그림 3. 문자가 분리 및 세션화된 영상
Fig. 3. Character separated and thinned image.

4. 특징점 추출 과정(9)

본 논문에서는 세션화된 문자 영상에서 특징점을 추출하여 이들로부터 문자를 인식하고자 한다. 한글 문자의 인식을 위해 필요한 특징점들은 최종점(terminal point), 가지점(branch point) 그리고 굴곡점(edge point)의 세가지 종류로 분류된다.

IV. 문자 인식의 과정

1. 모음 영역의 인식

한글문자를 인식하기 위해서는 한글문자를 자음과 모음으로 분리하여 인식하는 것이 효과적이다. 이 절에서는 문자 영상에서 모음의 형태를 알아내고 그 영역을 찾고자 한다. 모음의 형태와 영역을 찾기 위해서는 아래와 같은 단계를 따른다.

[단계 1] 모음의 시작 후보점을 찾는다. 모음의 시작 후보점은 횡모음은 1개 이하가 종모음에서는 2개 이하가 존재한다.

[단계 2] 모음의 시작 후보점에서 추적을 시작하여 모음에 연결된 교차점과 모음의 끝점을 찾는다. 만약 모음의 끝점이 존재하지 않으면 모음이 아니므로 다른 모음의 시작후보점을 찾는다.

[단계 3] 교차점이 존재하면 교차점에서 추적을 시작하여 모음의 끝점을 찾는다.

1) 횡모음 영역의 인식

횡모음 영역을 인식하기 위해서는 먼저 횡모음의 시작점을 찾고, 횡모음의 횡성분의 끝점을 찾아야한다. 횡모음의 시작 후보점이 존재할 수 있는 영역은 그림4와 같다. 그림4에서 빗금 친 영역내에 최종점이 그림5와 같은 방향으로 존재하면 이 점은 횡모음의 시작 후보점이 된다. 그리고 횡모음의 끝은 이 횡모음의 시작점과 연결된 횡성분의 가장 우측 끝을 말한다. 횡모음의 영역을 찾기 위해서는 횡모음 시작 후보점에서 추적을 시작한다. 횡모음 시작 후보점에서 그림5와 같은 방향으로 추적하여 최종점을 만날때까지 추적을 계속한다. 최종점을 만난다면 이 최종점은 추적을 시작한 횡모음의 횡성분 끝점이 된다. 그리고 횡모음에 존재할 수 있는 교차점의 형태

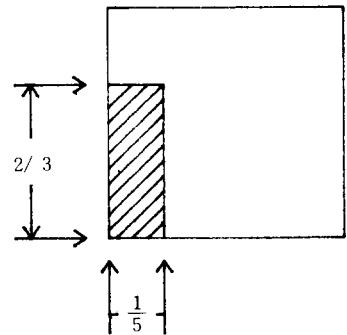


그림 4. 횡모음의 시작 영역
Fig. 4. Start region of horizontal vowel.

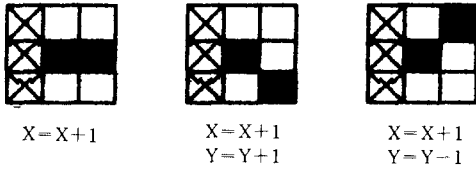


그림 5. 횡모음 추적의 방향
Fig. 5. Tracing direction of horizontal vowel.

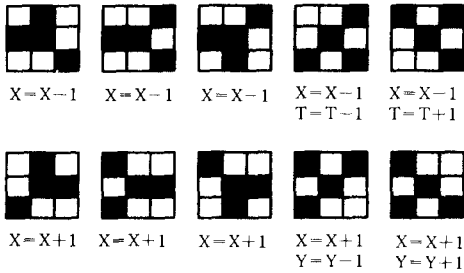


그림 6. 횡모음 교차점의 형태
Fig. 6. Intersection point type of horizontal vowel.

는 그림 6과 같다. 횡모음의 종성분은 이 교차점에서 추적하여 찾는다.

2) 종모음 영역의 인식

종모음 영역을 인식하기 위해서도 횡모음의 경우와 같이 먼저 종모음의 시작점을 찾고, 종모음의 종성분 끝점을 찾아야 한다. 종모음의 시작 후보점이 존재할 수 있는 영역은 그림 7과 같다. 그림 7에서 빗금 친 영역내에 최종점이 그림 8의 (a)와 같은 방향으로 존재하면 이 점은 종모음의 시작 후보점이 된다. 그리고 종모음의 끝은 이 종모음의 시작점과 연결된 종성분의 가장 하측 끝이다. 종모음에서 종성분의 끝을 만날때까지 추적하는 방향은 그림 8의 (b)이며, 교차점의 형태는 그림 9와 같다.

이상과 같은 방법으로 모음의 영역을 찾은 예가 그림 10과 같다. 그림 10에서 종모음의 시작점은 “V”, 횡모음의 시작점은 “H”로 표시되어 있다. 그리고 종모음의 끝점은 “#”, 횡모음의 끝점은 “*”로 표시되어 있다.

2. 자소 분리과정

자소 분리의 순서는 횡모음, 종모음, 초성자음, 종성자음의 순으로 분리한다.

1) 모음의 분리

모음은 시작점이 “H”혹은 “V”로 시작하며, 끝점은 “#”혹은 “*”로 끝난다. 따라서 모음의 분리는 모

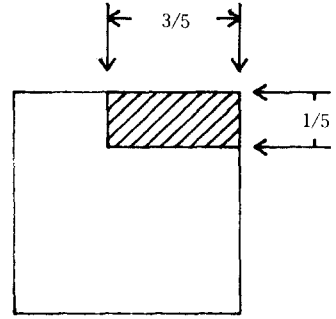


그림 7. 종모음의 시작 영역
Fig. 7. Start region of vertical vowel.

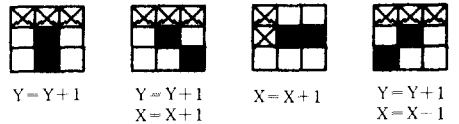


그림 8. 종모음 추적의 방향
Fig. 8. Tracing direction of vertical vowel.

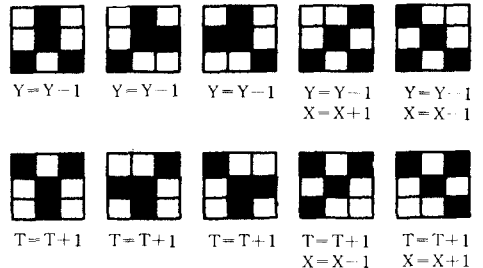


그림 9. 종모음 교차점의 형태
Fig. 9. Intersection point type of vertical vowel.

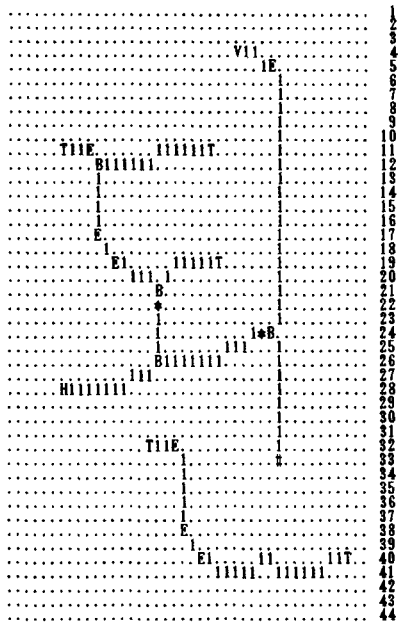
음의 시작점 “H”혹은 “V”와 연결성을 가진 모든 하소를 모음의 끝점 “#”혹은 “*”를 만날때까지 이동 시킴으로써 개별 문자에서 모음만 분리할 수 있다.

2) 자음의 분리

한글 문자에서 자음은 초성과 종성을 형성한다. 위의 알고리즘을 이용하여 모음을 분리해 내며 개별 문자 영역내에는 자음만 남게되고 이로부터 초성과 만약 존재한다면 종성을 분리해 내어 모든 자소의 분리를 끝내게 된다. 한글 문자의 기본형태에 따라 자음을 분리하기 위한 알고리즘은 다음과 같다.

(1) 기본 형태 : 초성+종모음

개별 문자 영역내에서 “H”는 없고 “V”만 존재하



* : 횡모음의 끝점 (Index point of horizontal vowel)
 # : 종모음의 끝점 (Index point of vertical vowel)

그림 10. 특징점 및 모음 영역 추출의 결과
 Fig. 10. Result of feature point and vowel region detection.

며 종모음의 최하단 y 좌표값이 문자높이의 90%보다 크면 이 문자는 초성과 종모음이 존재하는 경우가 된다. 이때 초성이 존재하는 위치는 종모음의 최대 x 좌표값 보다 왼쪽이 된다.

(2) 기본 형태 : 초성+횡모음

개별 문자 영역내에서 "V"는 없고 "H"만 존재하며 횡모음의 최대 y 좌표값이 문자높이의 80% 보다 크면 이 문자는 초성과 횡모음이 존재하는 경우가 된다. 이때 초성이 존재하는 위치는 횡모음의 최대 y 좌표값 보다 위쪽이 된다.

(3) 기본 형태 : 초성+종모음+중성

개별 문자 영역내에서 "H"는 없고 "V"만 존재하며 종모음의 최대 y 좌표값이 문자높이의 90% 보다 작으면 이 문자는 초성, 종모음, 그리고 중성이 존재하는 경우가 된다. 이때 초성과 중성의 위치는 y축에 대해 x축에 평행하게 스캔하여 화소가 존재하지 않는 y 축 좌표값을 찾는다. 이 좌표값의 위쪽은 초성, 아래쪽은 중성이 존재한다.

(4) 기본 형태 : 초성+횡모음+중성

개별 문자 영역내에서 "V"는 없고 "H"만 존재하며 횡모음의 최대 y 좌표값이 문자높이의 80% 보다 작으면 이 문자는 초성, 횡모음, 그리고 중성이 존재하는 경우가 된다. 초성과 중성의 위치는 기본 형

태 (c)에서의 알고리즘과 같다.

(5) 기본 형태 : 초성+종모음+횡모음

개별 문자 영역내에서 "H"와 "V"가 같이 존재하며 횡모음의 최대 y 좌표값이 문자 높이의 80% 보다 작으면 이 문자는 초성, 종모음, 그리고 횡모음이 존재하는 경우가 된다. 이때 초성이 존재하는 위치는 종모음의 최대 x 좌표값보다 왼쪽, 횡모음의 최대 y 좌표값보다 위쪽이 된다.

(6) 기본 형태 : 초성+종모음+횡모음+중성

개별 문자 영역내에서 "H"와 "V"가 같이 존재하며 횡모음의 최대 y 좌표값이 문자 높이의 80% 보다 크다면 이 문자는 초성, 종모음, 횡모음 그리고 중성이 존재하는 경우가 된다. 이때 초성과 중성의 위치는 기본 형태(c)에서의 알고리즘과 같다.

3. 자소 인식의 과정

앞절에서 개별 문자를 횡모음, 종모음, 초성자음, 중성자음의 순으로 분리하였다. 이 절에서는 분리된 각 자소를 인식하고자 한다. 본 논문에서는 분리된 자소를 인식하기 위하여 특징점의 위치와 형태 정보를 사용하였다. 먼저 모음은 횡모음과 종모음으로 분류한 후 모음에 존재하는 최종점과 가지점의 갯수를 이용하여 분류하면 표 1 과 같이 분류할 수 있다. 표 1에서 최종점과 가지점의 갯수가 같은 경우에는 횡모음에서는 중성분, 종모음에서는 횡성분 끝점의

표 1. 특징점에 의한 모음의 분류
 Table 1. Vowel classification using feature point.

모 음	특징점의 갯수		중/횡성분 끝점
	최종점	가지점	
횡 모 음	ㅡ		
	ㄱ	1	상측
	ㄲ	2	상측
	ㅋ	1	하측
종 모 음	ㆁ	2	
	㆏	1	우측
	㆑	2	우측
	㆓	1	좌측
중 모 음	㆕	2	좌측
	㆙	2	
	㆛	4	좌측
	㆝	1	좌측
기타	㆞	2	좌측
	㆟	2	좌측

방향과 위치를 비교하여 모음을 인식한다. 자음의 경우도 모음과 비슷하게 특징점을 분류하여 인식하였다.

V. 전처리용 신경회로의 설계

본 논문에서는 패턴 분류를 위해 하드웨어 구현이 쉬운 퍼셉트론을 응용하였다. 퍼셉트론은 1959년 Rosenblatt에 의해서 제안된 모델로서 그림11과 같은 노드들간의 연결구조를 갖고 있다. 퍼셉트론의 주기능은 패턴 분류로서 입력 패턴을 구별하여 그 패턴이 어느 분류에 속하는가를 분류하는 것이다. 이러한 단층 퍼셉트론의 알고리즘을 이용하여 문자 인식의 전처리 과정에 응용하였다.

생물학적 뉴론은 정보를 전달할때 흥분과 억제 신호를 사용하며, 뉴론과 뉴론의 연결 상태, 즉 시냅스로써 정보를 기억한다. 흥분 신호는 다른 뉴론의 활동을 여기시키고 억제 신호는 다른 뉴론의 활동을 저지하는 역할을 한다. 이러한 흥분과 억제 신호를 PMOS와 NMOS를 이용하여 신경회로망으로 구현할 수 있다. 그림12는 신경회로망을 형태 분류기에 응용한 예이다. 그림12(a)의 마스크를 CMOS 회로로 구현하면 (b)와 같은 회로가 된다. 그림12(b)의 형태 분류기는 그림11의 single layer perceptron model을 응용한 것이다. 그림12(a)의 3*3 mask pattern에서 input BIT 1=BIT 4=BIT 5=1이고 BIT 2=BIT 3=BIT 6=BIT 9=0이다. 그리고 input BIT 7, BIT 8은 don't care이다. 이 3*3 mask pattern을 그림11의 single layer perceptron model로 구현하려면 input의 수는 9개가 되어야 하고, output수는 3*3 mask pattern의 수가 되어야 할 것이다. 우선 input으로는 그림12(a)의 3*3 mask pattern을 사용하고,

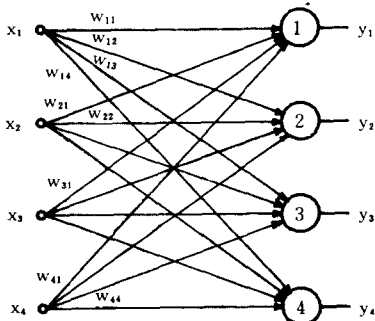


그림 11. 단층 퍼셉트론 모델 Fig. 11. Single layer peceptron model.

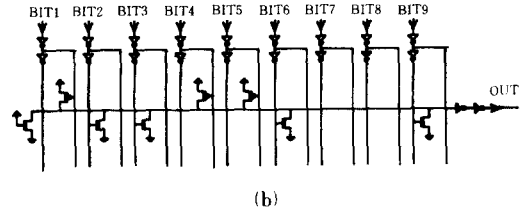
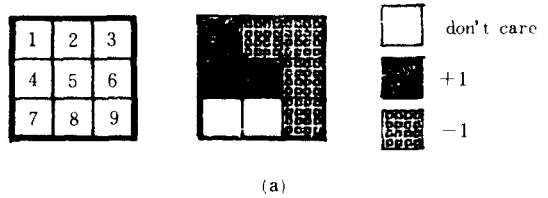


그림 12. 단층 구조 퍼셉트론의 형태 분류기에의 응용 (a) 3*3 마스크 패턴의 예 (b) 형태 분류기의 CMOS 회로

Fig. 12. Single layer perceptron for pattern classifier.

(a) Example of 3*3 mask pattern, (b) CMOS circuit of pattern classifier.

output 수를 4개로 가정하고 설명하면 그림12(a)의 3*3 mask pattern에서 input X1=X5=1이고, X2=X3=X6=X9=0, 그리고 X7, X8은 don't care이다. 이런 input이 들어왔을때 output y1만이 high, 나머지 output y2, y3, y4를 low가 되게 하려면 weight 값을 W11=W41=W51=1로, W12=W13=W14=W21=W22=W23=W24=W31=W32=W33=W34=W42=W43=W44=W52=W53=W54=W61=W62=W63=W64=W91=W92=W93=W94=-1, 그리고 W71=W72=W73=W74=W81=W82=W83=W84=0로 해서 output y1의 값만 1(>threshold 값)로, 나머지 output y2, y3, y4는 (0<threshold 값)로 만든다. 회로에서 PMOS는 VDD, 즉 흥분신호를 두 개의 인버터로 구성된 뉴론에 전달하고 NMOS는 GROUND, 즉 억제 신호를 뉴론에 전달한다. 뉴론에서는 전달된 흥분과 억제 신호의 합이 뉴론의 문턱치 값보다 크면 "High"의 값을, 작으면 "Low"의 값을 유지한다. 이러한 뉴론의 동작을 신경회로망에 응용하면 다음과 같다. 마스크에서 +1의 상태는 PMOS로, -1의 상태는 NMOS로, 0의 상태는 연결되지 않는 것으로 하여 형태 분류기의 연결상태를 결정하였다. 그리고 PMOS의 W/L 값은 6μm/2μm로, NMOS의 W/L 값은 2μm/2μm로 하여 컨택턴스의 값을 같게 하였다. 또한 마스크 형태와 동일한 입력에 의해서만 뉴론의 출력이 "High"가 되도록 하기 위해서 입력과

관계없는 NMOS 소자를 하나더 추가하였다. 이 NMOS의 W/L 값은 $(2 \cdot N) \mu\text{m} / 2 \mu\text{m}$ 로 주어진다. 여기서 N은 마스크에서 결정된 +1 상태의 갯수이다. 그리고 공정변수에서 전자의 이동도는 $580(\text{cm}^2/\text{V}\cdot\text{sec})$ 이고, 정공의 이동도는 $200(\text{cm}^2/\text{V}\cdot\text{sec})$ 이다. 이렇게 함으로써 형태 분류기의 9개의 입력이 마스크와 동일하다면 뉴론은 "High"의 상태를 유지하고 하나의 입력이라도 다르다면 "Low"의 상태를 유지하도록 하였다. 그림 13은 잡음을 제거하기 위한 5×5 마스크이다. 문자 영상에서 5×5 마스크 형태로 25개의 화소를 입력받는다. 입력된 화소의 형태가 그림 13의 잡음 제거 마스크와 일치하는 경우가 있다면 잡음성분이 존재하는 것이다. 이렇게 잡음 성분이 존재하면 마스크의 중앙부분과 일치하는 문자 영상의 화소를 지운다. 회로설계는 본문의 그림 12(b)의 형태 분류기 회로와 같은 방법으로 하였다. Input 수는 25개이고, output 수는 9개이다. 그림 14는 단층 구조 퍼셉트론으로 구현한 세선화 회로이다.

VI. 결과 및 고찰

본 논문에서는 스캐너로 부터 입력된 한글 영상에서 특징점을 추출하여 소프트웨어로 한글 문자를 인식하였다. 이러한 소프트웨어에 의존하는 문자 인식 시스템이 가지는 인식 속도의 한계를 극복하기 전처리 과정을 신경 회로망의 한 모델인 단층구조 퍼셉트론 모델로 설계하였다. 한글 문자의 자소를 분리하는데 많은 어려움이 따라 인식률은 낮았다. 앞으로

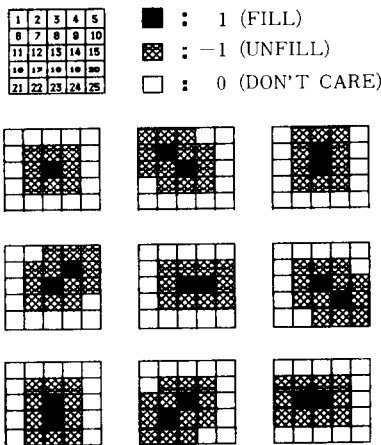


그림 13. 잡음 제거를 위한 마스크 형태
Fig. 13. Mask pattern for noise elimination.

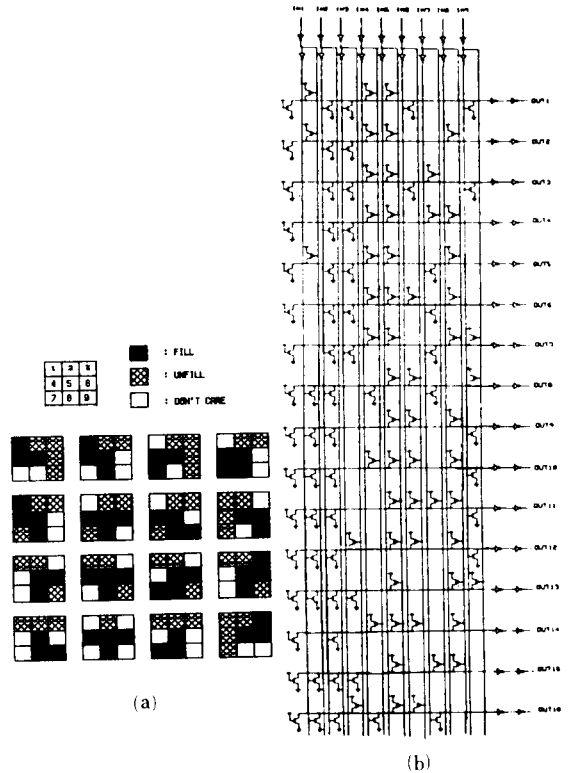


그림 14. 세선화 회로의 설계
(a) 세선화 마스크
(b) 세선화 회로

Fig. 14. Design of the thinning circuit.
(a) Thinning mask,
(b) Thinning circuit.

한글 문자의 자소분리에 대한 연구가 더욱 필요하겠

VII. 결 론

본 논문에서는 스캐너에서 입력된 한글 영상으로부터 특징점을 추출하고 이를 이용하여 한글 문자를 인식하였으며, 그 전처리용 신경 칩을 설계하였다. 한글 문자의 크기가 45×45 화소인 경우 한개의 한글 문자를 인식하는데 소요된 시간은 약 3.5초 정도 이었는데, 전처리 과정에서 전체의 약 65%인 2.2초 가량이 소요되었다. 이러한 소프트웨어에 의존하는 문자 인식 시스템이 가지는 인식 속도의 한계를 극복하기 위해 잡음제거, 선형화, 세선화, 직선화 및 특징점 추출등의 전처리 과정을 단층 구조 퍼셉트론의 개념으로 설계하였다. 설계된 회로는 SPICE를 사용하여 시뮬레이션하였으며 이중급속 CMOS $2 \mu\text{m}$

향 총 장 에 써 및 라 늬 게
 합 초 작 었 성 민 등 늬 것
 합 찻 자 없 섰 미 듯 노 검
 할 처 지 었 업 설 물 들 년 결
 한 체 입 어 선 문 든 너 건
 학 청 임 양 석 무 드 내 거
 하 철 일 약 서 못 두 남 개
 피 천 인 야 생 목 될 남 까
 표 처 이 앞 새 모 된 난 갈
 경 차 의 앓 상 명 되 나 강
 편 집 음 알 산 면 또 김 갔
 관 질 음 암 사 며 동 길 값
 과 직 은 아 뿐 어 도 끝 같
 티 지 은 아 뿐 매 데 굵 간
 특 중 유 심 본 말 떠 금 각
 트 주 위 심 부 만 던 그

향 총 장 에 써 및 라 늬 게
 합 초 작 었 성 민 등 늬 것
 합 찻 자 없 섰 미 듯 노 검
 할 처 지 었 업 설 물 들 년 결
 한 체 입 어 선 문 든 너 건
 학 청 임 양 석 무 드 내 거
 하 철 일 약 서 못 두 남 개
 피 천 인 야 생 목 될 남 까
 표 처 이 앞 새 모 된 난 갈
 경 차 의 앓 상 명 되 나 강
 편 집 음 알 산 면 또 김 갔
 관 질 음 암 사 며 동 길 값
 과 직 은 아 뿐 어 도 끝 같
 티 지 은 아 뿐 매 데 굵 간
 특 중 유 심 본 말 떠 금 각
 트 주 위 심 부 만 던 그

(a)

(b)

그림 15. 인식의 예

- (a) 입력 영상
- (b) 인식된 결과

Fig. 15. Example of recognition.

- (a) input character image,
- (b) recognized result.

설계 규칙으로 KUIC-LED 상에서 레이아웃 하였다.

한글 자소의 분리 및 인식에 대한 실험은 중학교 국어 교과서 및 한글 배열표에서 사용 빈도수가 높은 350자의 한글 문자에 대해 수행하였다. 그 결과 한글 문자의 자소 분리율은 약 97%였으며 인식율은 약 95%였다. 그림15는 인식의 예이다. 인식율 95%는 전 과정을 소프트웨어로 시뮬레이션한 결과이며, 전처리 과정을 chip으로 구현하면 처리 속도는 개선될 것이 기대되나 인식률과는 무관하다. 전처리 과정의 소프트웨어 수행시간은 2.2초 정도이다. 이 부분을 chip화 하였을 경우 한 문자당 인식시간은 약 1.3

초 정도가 기대된다. 본 논문에서는 필기체 한글 문자에 대해서는 인식하지 못하였으며 전처리 과정의 회로만 설계되었다. 앞으로는 필기체 한글문자의 인식과 함께 특징점 분류 방법을 다층 구조 퍼셉트론 개념으로 설계하여 한글 문자의 전 인식과정을 하드웨어로 구현하는 연구가 필요하겠다.

參 考 文 獻

- [1] 남기곤, 박의열, 이양성, "겹쳐진 물체의 인식을 위한 정합 알고리즘," 전자공학회 하계종합학술대회 논문집, pp. 671-674, 1988.
- [2] 권중장, "특징점을 이용한 한글 문자의 인식," 한국과학기술원 석사학위 논문, 1987.
- [3] 이주근, 남궁재찬, 김영진, "한글 패턴에서 subpattern 분리와 인식에 관한 연구,"전자공학회지, pp. 1-8, June, 1981.
- [4] 고병기, 김진형, 양현승, "경쟁하는 신경망 조직을 이용한 인쇄체 한글문자의 인식,"제1회 영상처리 및 이해 워크샵 논문집, pp. 127-134, 1989.
- [5] K. Fukusima, "A neural network for visual pattern recognition," *IEEE Computer Magazine*, pp. 65-75, 1988.
- [6] J.J. Hopfield, "Artificial neural network," *IEEE Circuits and Device Magazine*, pp. 3-9, 1988.
- [7] 배창석, "신경회로망 알고리즘을 이용한 한글 문자 인식," 경북대학교 석사학위 논문, 1989.
- [8] 류제광, "APOLLO workstation 상에서 Layout Editor (KUIC-LED)의 이식,"한국통신학회, 추계학술발표회 논문집, pp. 353-355, 1989.
- [9] 김종렬, 정호선, 이우일, "신경 회로망의 개념을 이용한 문자 두께변환 회로의 설계," 전자공학회, 추계종합학술대회논문, pp. 520-522, 1988.

著 者 紹 介

金 琮 烈 (正會員) 현재 한국전기통신공사 근무 중

李 宇 一 (正會員) 第26卷 第5號 參照
현재 경북대학교 전자공학과 교수

鄭 鎬 宣 (正會員) 第26卷 第5號 參照
현재 경북대학교 전자공학과 부교수