

분절 화소들의 특징을 이용한 필기체 숫자인식

최용호^o 조범준

조선대학교 전자정보공과대학 컴퓨터공학부
pcanuts@stmail.chosun.ac.kr, bjcho@mail.chosun.ac.kr

Handwritten Numeral Recognition using the Features of Segmented Pixels

Yong-Ho Cho^o Beom-Joon Cho
Dept. School of Computer Engineering, Chosun University

요 약

필기체 숫자 인식을 위한 새로운 특징 추출방법을 숫자의 기하학적인 구조들을 이용하여 연구 제안하였다. 일반적으로 쓰이고 있는 특징점들의 몇가지 부류를 결정하여 추출하였고, 분절 화소들을 이용한 특징 추출기는 사소한 부분들을 명확한 특징으로 탐지하여 추출하게 된다. 신경망은 새로운 접근 가능성을 탐지하는 실험 인식기로 사용하였고, 이러한 방법들을 이용하여, 일반적인 특징점 추출방법과 본 연구에서 제안하는 특징점 추출방법을 결합하게 되면 필기체 문자의 인식이 단순히 일반적인 특징만을 활용하여 얻는 인식률 보다 훨씬 향상됨을 보여주었다.

1. 서 론

필기체 숫자 인식 방법은 사용 빈도수가 가장 많은 부분으로 꾸준히 연구가 되어지는데 이는 수표나, 신용카드 매출전표, 세금계산서, 각종 전표 및 문서 등은 작업의 자동화와 전자 문서화를 지향하여 검색, 수정이 용이하며 업무 처리 속도 등을 향상 시킬수있다. 또한 이들은 모두 금액을 다루는 경우가 많기 때문에 높은 인식률과 신뢰도를 필요로 하기 때문이다. 그래서 이 분야는 상당한 진보를 해왔고, 많은 숫자인식 시스템들이 상품화 되었지만 인식률이 떨어지는 문제를 안고 있다. 필기 형태의 구애를 받지 않고 어떤 형태로든 인식 되는 것에 더욱더 연구할 여지가 남아있다.

일반적으로 필기체 숫자인식 시스템은 크게 3가지 요소로 구성 되어지는데, 전처리와 특징추출, 그리고 분류(인식)으로 나눌 수 있다. 이중 특징추출은 최종 인식 성능에 영향을 미치는 주요 요인으로서, 필기체 숫자 인식에서 매우 중요한 역할을 한다. 그래서 아직도 많은 특징 추출방법이 연구 되어지고 있는데, [1,2] 본 연구 논문에서는 필기체 숫자를 기하학적인 면에서 특징추출 방법을 연구하였다.

제 2장에서는 2가지 특징 추출방법에 대해서 보여준다. 일반적인 특징 추출 방법과 분절 화소들의 특징 탐색기는 필기체 숫자들의 조각난 부분들을 8가지 방법으로 탐색하도록 되어있다. 3장에서는 오류역전파(Error Backpropagation)알고리즘을 이용하여 학습하였다. 제4장에서는 시스템의 성능을 실험 하였고, 마지막은 결론 및 고찰로 맺는다.

2. 특징 추출

대부분 숫자에 대한 이상적인 형태는 연결된 획들이 연속적 이었을 때 이다. 잡음 제거나 분리, 일반화 과정인 전처리 단계가 필요하며 분리된 숫자 연결과 문자의 비틀림 등을 교정하는 알고리즘을 이용하여 특징 추출에 명확성을 더 하였고[3], 전처리 과정이 끝난 숫자는 위치를 바꾸지 않고도 32*24로 만들어 진다.

본 연구 시스템에서는 발표된 연구자료를 바탕으로 제안하게 될 2가지 유형의 특징 추출을 참고하였다.

2.1 일반적인 특징 추출

2.1.1 오목 점 특징

오목 점 특징이라는 것은 윗면/아랫면/왼쪽/오른쪽의 관점으로 볼 때 각각 숫자의 바깥쪽 윤곽선이 오목임을 나타낸다. 윤곽선 방향 i 번째는, j 번째 오목 점 특징에 대해서 처리 되어진다.

$$C_{ij} = d_{ij} / e_{ij} \quad (1)$$

d_{ij} 가 오목 점의 곡선 한 가운데 있다면, e_{ij} 는 그 오목 점 곡선의 넓이를 말한다. 시스템에서는 $c_{ij} > 0.3$ 이면, 오목 점 특징을 고려하게 된다. 하지만 반대에 경우라면 오목 점 특징은 무효한 것으로 처리된다. 각 윤곽선 오목 점의 갯수와 요면의 위치는 특징 점으로 사용된다.

그림1의 숫자 "2"에 대해서 왼쪽 면의 상위는 1개의 오목 점이 존재하고, 오른쪽 면의 하단에는 오른쪽 오목 점이 존재하고, 또 아랫면 하단에도 오목 점이 있다.

숫자 “8”에 대해서는 양쪽 수직면의 중간 부분에 왼쪽/오른쪽 오목 점 만이 존재한다.

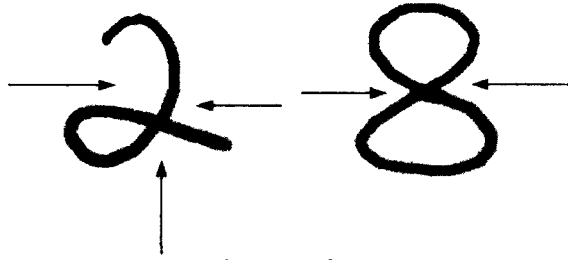


그림.1 오목 점 특징

2.1.2 넓이 특징

숫자는 4개의 영역으로 수직 방향으로 나누어진다. 각 하위 영역의 최대 넓이가 계산 되어지면 m_1, m_2, m_3, m_4 로 표기된다.

하위 영역 x 의 넓이를 측정하기 위해서는 크기를 조정하는 함수 $f(x)$ 는 다음과 수식과 같이 계산된다.

$$f(x) = \text{int} \left(\frac{\alpha^* (x - M_{\min})}{(M_{\max} - M_{\min})} \right) \quad (2)$$

$M_{\max} = \max\{m_i\}, M_{\min} = \min\{m_i\}, i = 1, \dots, 4$ 라면 α 는 3과 같아지도록 크기가 조정된다. 넓이 측정 다음에 $f(x)$ 는 정수이며 2비트로 특징화 된다.

넓이 특징을 이용함으로써 하위 부분의 넓이 비율을 비교함으로써 “1”과 “7”을 그리고 “0”과 “8”을 쉽게 판별할 수 있다.

2.2 분절 화소들의 특징 추출

위에서 언급된 추출 방법들은 일반적인 특징점으로 인식 할 수 있으나, 변형이 심한 조건에서는 필기체 숫자를 효과적으로 인식할 수 없다. 따라서 분절 화소들의 특징 탐색기를 이용하여 어떠한 문자 모양에서 조그마한 부분들도 특징점으로 추출하고자 한다.

SPFD(w, h)는 2개의 파라미터인 w 와 h 를 가진 탐색기이며, 이것은 특징(분절)점을 찾게 된다. SPFD(w, h)는 특징점이 탐색 되어지면 1이 부과 되고, 그렇지 않으면 SPFD(w, h)=0이 부과 된다. 다양한 방향으로 분절들을 찾기 위해서 그림2에 보이는 것과 같이 8개의 방향이 도입되어 있다. 각각은 특정한 방향으로 향해있는 분절들을 추출하기 위해 사용된다.

탐지기 (1)은 왼쪽 면에서 수평적인 분절을 찾는데 사용되어지고, 탐지기 방향은 하단에서 상단으로 움직인다. 탐지기(2) 역시 왼쪽면에서 수평적인 분절을 찾게 된다. 하지만 탐지기의 방향은 맨 상단에서 하단부로 움직인다.

탐지기 (3)과 (4)는 둘 다 오른쪽 면에서 수평적인 분절을 탐지 할 수 있다. 탐지기(5)와 (6)은 상단 면에서 수직적인 분절을 찾는데 사용되어 지고, 탐지기(7)과 (8)은 하단면에서 수직적인 분절을 찾는데 사용된다.

그리고 이들 탐지기의 움직이는 방향은 그림2에서 볼 수 있다.

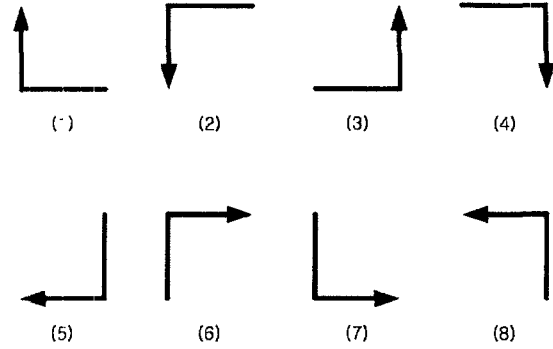


그림.2 8방향 탐지기

그림.2 (1)의 탐지기를 설명하면 형질의 왼쪽 면을 따라 수평적인 분절을 찾을 수 있다. 예를 들어 P_0 화소의 높이와 P_0 화소의 넓이를 지닌 “2”를 그림.3에서 볼수 있는데, 상단의 왼쪽 가장자리에 수평적인 작은 특징(분절)점이 있다. 탐지기의 움직이는 방향은 위치 1, 2, 3에서 볼수 있다. 위치 2는 다른 것들 보다 더 수평적인 분절을 찾을 수 있다.

SPFD(w, h)에서 h 는 높이를 말하며, w 는 넓이를 나타내기 때문에 훨씬 더 중요한 파라미터가 될 것이다. W 가 크게 고정 시킨다면 탐지기는 많은 특징(분절)점을 간과 시켜버릴 것이고, 너무 작게 고정되면 지그재그 꼴의 잡음과 난월들과 같은 많은 사항들이 탐지되어 동일한 문자에 대해 특징점이 변화 될 것이다. 파라미터(w_0, h_0)쌍이 적절하려면 탐지기가 사용되기 전에 초기화 되어야 한다. 본 시스템에서는 w_0 와 h_0 는 몇가지 실험적 결과를 바탕으로 숫자의 높이와 넓이를 1/8에 각각 할당하였다.

그림.2(1) 탐지기를 이용하여 특징점을 구하는 알고리즘은 다음과 같다.

- 1단계 : 탐지기를 숫자의 왼쪽 하단부 가장자리에 위치시킨다.
- 2단계 : w_0, h_0 를 초기화 시킨다.
- 3단계 : 분절을 탐지한다.
- 4단계 : (SPFD(w_0, h_0)=1) 분절이 탐지되면 다음 단계로 간다
- 5단계 : 탐지기가 맨 상단에 도달했는지 확인. 도달했다면 8단계로 가고, 그렇지 않으면 3단계로 되돌아간다.
- 6단계 : 탐지기를 멈추고 w_i 를 증가, h_i 를 다음과 같이 탐지한다.


```
While (FFD( $w_i, h_i$ )=1) {
                 $w_{i+1} = w_i + 1$ ; detect  $h_{i+1}$ 
            }
```
- (w_n, h_n)파라미터 값을 적는다
- 7단계 : 탐지된 분절점으로 이동하여, 탐지기가 맨 상단에 도달했는지의 여부를 확인한다. 도달하지 않으면 2단계로 간다.
- 8단계 : 탐색 종료

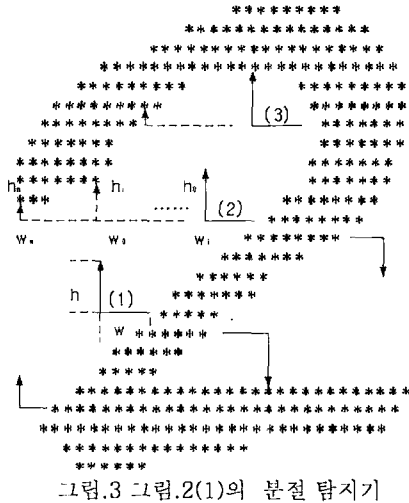


그림.3 그림.2(1)의 분절 탐지기

숫자 "2"는 그림2(1)을 이용하여 상단 왼쪽의 부분을 추출, 그림2(2)는 하단 왼쪽 부분, 그리고 그림2(4)는 하단 오른쪽 부분을 추출 한다. 숫자 "5"에 대해서는 그림2(3)을 이용 상단 오른쪽 부분을 탐지하고, 그림2(2)는 하단 왼쪽 부분을 탐지한다. 그래서 왼쪽 면에서 수평적 특징들이 존재하는 것들은 "2", "3", "4", "5", "7", "8", "9"등이고, "2", "3", "4", "5", "6", "7", "8"과 같은 숫자들은 오른쪽 면에서 특징들이 존재한다. 제안된 탐색기를 활용하면, 필기체 숫자 "4"와 "9", "5"와 "3", 그리고 "2"와 "3"은 쉽게 구분된다.

3. 신경망 인식 시스템

숫자를 인식하기 위하여 3개층 구조를 갖는 클러스터 MLP 신경망을 사용하였다. 본 연구에서 제안된 구조는, 입력층은 3개의 7*5 유니트로 구성되어 국부적인 은닉 유니트와 꺾짐을 이루어 완전 연결 되었다[4]. 은닉 유니트는 2개의 7*5 유니트로 구성 되어지고, 출력층은 모두 10개의 유니트로 구성되며 은닉 유니트 출력 값에 의해 각 부류의 숫자를 분류하도록 학습된다. 출력층에 있는 10개의 유니트는 입력 부류에 해당하는 유니트만이 1값을 갖고 그 외의 유니트는 0값을 갖도록 하였다.

예를 들어 "7"에 대한 정규화 영상이 입력되었을 때, "7"을 나타내는 유니트만이 1값을 갖고 그 외의 유니트는 0이 된다.

학습 알고리즘은 오류 역전파(BP)알고리즘을 사용하였으며, 유니트의 출력 함수로는 다음과 같은 시그모이드 함수를 사용한다.

$$f(o) = \frac{1}{1 + e^{-(\sum WX + bias)}} \quad (3)$$

여기서 o는 비선형 출력함수의 출력값이며, W는 이전 노

드로 부터의 가중치이고 X는 입력 값이다. W와 bias는 초기에 임의로 설정하였으며 학습 과정을 통하여 갱신되도록 하였다. 가중치의 갱신은 일반화된 델타법칙을 이용하였으며 다음과 같은 식에 의하여 이루어진다.

$$\Delta w(n) = -\partial E / \partial w + \alpha \Delta w(n-1) \quad (4)$$

n은 학습 횟수를 나타내고, α는 모멘텀을 나타낸다. 그리고 E는 기대값과 차이값을 제곱한 값이다.

4. 결론 및 고찰

본 연구에서 사용된 필기체 숫자 데이터는 조선대학교 학생을 대상으로 250명에게 (가로)2.3cm×(세로)2.5cm 10개의 사각형 안의 각각에 제안을 두지않고 0부터 9까지의 숫자를 각각 2회씩 수정 사인펜으로 쓴 숫자 5000자를 이용하였으며, 그 중 2500자를 학습 데이터로 사용하였으며, 2500자는 최종 실험 데이터로 사용하였다.

	0	1	2	3	4	5	6	7	8	9	정인식 (%)	오류 (%)	기각 (%)
0	239	0	0	1	0	1	3	1	0	0	239 (95.60%)	5 (2.00%)	6 (2.40%)
1	0	245	0	0	0	0	0	1	0	0	245 (98.00%)	1 (0.40%)	4 (1.60%)
2	0	0	238	1	0	0	3	0	1	0	238 (95.20%)	5 (2.00%)	7 (2.80%)
3	0	0	2	240	0	0	3	0	2	0	240 (96.00%)	7 (2.80%)	3 (1.20%)
4	2	0	1	0	239	0	1	0	0	0	239 (95.60%)	4 (1.60%)	7 (2.80%)
5	0	1	0	2	0	238	4	0	0	0	238 (95.20%)	7 (2.80%)	5 (2.00%)
6	3	2	1	1	0	0	235	0	3	0	235 (94.00%)	10 (4.00%)	5 (2.00%)
7	1	3	2	0	1	0	0	241	0	0	241 (96.40%)	7 (2.80%)	2 (0.80%)
8	3	1	0	1	0	0	2	0	235	5	235 (94.00%)	12 (4.80%)	3 (1.20%)
9	2	0	0	2	2	1	0	5	1	232	232 (92.80%)	11 (4.40%)	7 (2.80%)
Σ											2382 (95.28%)	69 (2.76%)	48 (1.96%)

표.1 인식률(%)

실험 결과는 위의 표.1에 나타나 있으며, 구조적인 특징만을 사용하였을 때보다 약 3%정도의 인식률이 높아졌으며, 오류율도 낮아지는 결과를 얻게 되었다.

5. 참고문헌

[1] Y.C Chim, A.A.Kassim, and Y.Ibrahim. Character Recognition Using Statistical Moments. Image and Vision Computing, 1999, pp299-307.
 [2] O.D.Trier, Anil K.Jain and Torfinn Taxt. Feature Extraction Methods for Character Recognition-A Survey. Pattern recognition, 29(4) 1996, pp641-662.
 [3] Jinhai Cai and Zhi-Qiang Liu. Integration of structural and Statistical Information for Unconstrained Handwritten Numerical Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, 21(3) 1999, pp263-270.
 [4] Y. le Cun et al, "Constrained Neural Network for Unconstrained Handwritten Digit Recognition," Proc. 1st Int. Workshop on Frontiers in Handwriting Recognition, Montreal, Canada, Apr. 1990, pp. 145-154.