자소분할과 픽셀분포를 이용한 한글문자인식

조영국, 이동욱
동국대학교 전기공학과

Recognition of Hangeul Character Using Grapheme Segmentation and Pixel Distribution

Young-Guk Cho, Dong-Wook Lee
Dept. of Electrical Engineering, Dongguk University

Abstract - 한글 문자 인식에 관한 연구는 통계적 방법과 구조적 방법, 신경 회로망 등 다양한 방법론이 제시되어 왔다. 그러나 한글은 영문이나 숫자에 비해 방대한 문자수와 복잡한 구조로 인하여 인식에 많은 어려움을 가지고 있다. 따라서 본 논문에서는 한글을 가장 단순한 구조인 자음과 모음으로 분리한 뒤 각 개체의 픽셀 분포를 파악하고, 한글의 구조적 특징을 이용하여 자소의 행과 열에서의 peak값과 픽셀의 분포를 그룹으로 나누어 한글을 인식하는 방법을 제시한다.

1. 서 론

문자 인식 시스템은 그 핵심요소 기술에 따라 통계적인 방법과 구조적인 방법, 신경망에 의한 방법 등 여러 가지 방법이 시도되고 있으며, 인식하려고 하는 대상에 따라 각각 장점과 단점을 가지고 있다.
구조적인 방법은 문자패턴의 구조적인 정보를 이용하여 인식을 행하는 방법으로 문자패턴의 구조적인 관련성을 추출하기가 쉽기 때문에 많이 이용되며 문자의 본질적인 특성을 이용하여 인식하므로 다른 방법들에 비해 오인식이 작다는 장점을 지니고 있다. 또한 구조적인 방법은 이론적으로 단순하고 명백하기 때문에 확장이나 수정 보완이 편리하다. 그러나 필기체 문자 패턴과 같은 다양한 패턴에 대해서는 많은 규칙들이 필요하며 이로 인해 인식기간이 길어지는 단점들이 있다.
본 연구에서는 이러한 문제점을 개선하기 위해 한글을 가장 기본적 단위인 자음과 모음으로 분할한 뒤 각각 개체들의 행과 열에서의 픽셀 분포도와 peak값을 구하여 이들 구조적 특징 데이터를 K-means 알고리즘을 이용하여 분류하는 방식을 제안한다.

2. 본 론

2.1 한글 문자의 특성

한글 문자는 기본적으로 자음 14개와 모음 10개로 구성되며 이들의 다양한 조합 규칙에 의해 최소 2개에서 최대 6개의 기본 자소가 모여 하나의 의미있는 문자를 이루는데, 이러한 조합에 의해 총 11,172자의 한글문자 조합이 가능하다.
한글 문자의 구조적 특징을 살펴보면 다음과 같다.
첫째, 한글 문자의 외형은 사각형의 틀 안에 들어갈 수 있다.
둘째, 한글 문자의 자소들은 대부분 직선의 패턴이 추가되어 하나의 자소가 형성되는데 주직선의 위치는 주로 행과 열의 각각 양끝과 중앙의 3곳에 위치하며 각각의 자소들은 고유의 peak값을 가진다.
셋째, 유사한 자소들이 많아 오인식의 원인이 된다.
(예, "ㄱ"과 "ㅋ", "ㅏ"와 "ㅑ"등)
이러한 특징들을 이용하여 행과 열에서의 픽셀의 분포도를 조사하여 보면 <그림 1>과<그림 2>와 같이 각각 다른 픽셀의 분포도를 나타내게 된다.

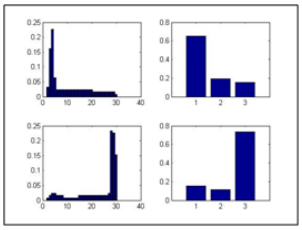

그림 1. 'ㄱ'의 픽셀 분포도

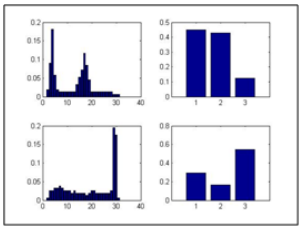

그림 2. 'ㅋ'의 픽셀 분포도

그림에서 왼쪽의 그래프들은 '행'과 '열"에서의 픽셀값을 나타내며, 오른쪽 그래프들은 왼쪽·중앙·오른쪽 3그룹으로 묶어서 분포도를 계산한 값이다. <그림.1>에서 "ㄱ"의 그룹 분포도는 "행"에서 [60% 30% 10%], "열"에서 [38% 35% 27%]의 비율로 분포되어 있으며, '행'에서는 1번, '열'에서도 1번의 peak가 발생했음을 알 수 있다. 반면, <그림.2>에서 "ㅋ"의 그룹 분포도는 "행"에서 [60% 30% 10%], "열"에서 [38% 35% 27%]의 비율로 분포되어 있으며, '행'에서는 2번, '열'에서 1번의 peak가 발생했음을 알 수 있다. 이렇게 비슷한 개체들의 대해서도 주직선의 개수와 분포의 위치에 따라서 발생하는 행과 열에서의 3그룹의 분포도와 peak값의 8개의 특징 데이터들을 비교하면 "ㄱ"와 "ㅋ" 뿐만 아니라 "ㅏ"와 "ㅑ" 같이 유사한 자소들도 명확하게 구분되어질 수 있다.

2.1.1 윤곽선 추적을 통한 개체분리

문자영상을 인식하기 위해서는 문자의 개체들을 정확하게 추출하는 것이 중요하다. 본 논문에서는 문자의 개체들을 분리하기 위해 "4방향 윤곽선 추적 알고리즘(김성영외, 1999)"[1]을 적용한다.

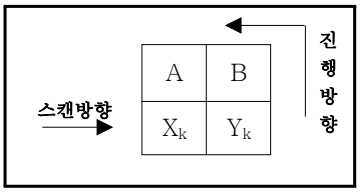

그림 3 윤곽선 추출을 위한 2*2 마스크

A	B	X_{k+1}	Y_{k+1}	결 과
1	0	A	B	전 진
0	1	B	Yk	Y고정, 시계방향 회전
1	1	A	Xk	제자리 시계방향 회전
0	0	Xk	A	X고정, 반시계방향 회전

표 1. 마스크의 A, B에 따른 진행방향

4방향 윤곽선 추적 알고리즘은 <그림3>의 2*2 마스크를 이용하여 왼쪽위부터 스캔하다가 X_k가 개체의 윤곽선과 처음 만나는 점을 시작점으로 하여 A와 B에 대응하는 두 픽셀을 고려하여 마스크 진행 방향을 결정하게 되며, Xk가 지나간 자리가 영상의 윤곽선이 된다. 마스크 진행 방향은 A와 B가 모두 배경일 경우에는 Xk를 기준으로 반시계방향으로 회전하고, A가 윤곽선이고 B가 배경일 경우에는 Xk가 A로 이동하면서 마스크는 한 픽셀 앞으로 전진한다. 또한 A가 배경이고 B가 윤곽선일 경우에는 마스크는 Yk를 기준으로 시계방향으로 회전하고 A와 B가 모두 윤곽선일 경우는 Xk는 Yk는 Xk로 이동하여 제자리에서 시계방향으로 회전하게 된다. 다음과 같은 규칙으로 윤곽선을 따라 이동하다가 마스크의 Xk가 시작점으로 돌아오게 되면 개체를 저장 후 윤곽선 안쪽의 이미지를 삭제하고 다음 개체의 윤곽선을 스캔하기 위해 이동한다. 윤곽선 추적 알고리즘을 이용하여 개체의 크기에 따라 잡음제거도 할 수 있다. <표1>는 A와 B의 값에 따른 Xk와 Yk의 진행방향을 나타내었다.

2.2 K-means 알고리즘을 이용한 분류

K-means (MacQueen, 1967) [2]은 유명한 군집화 문제를 해결하는 가장 간단한 자율학습(Unsupervised Learning)알고리즘 중 하나이다. 사전

에 정해진 어떤수의 클러스터를 통해서 주어진 데이터 집합을 분류하는 간단하고 쉬운 방법이다.

4방향 윤곽선 추적 알고리즘에 의해 확인된 자소들을 수직, 수평 방향으로 투영하여 여백에 해당하는 백화소는 제거하고 문자를 구성하는 흑화소의 개수의 합을 구한다. 그리고 문자의 두께에 따라 픽셀의 개수가 다르기 때문에 이를 맞춰주기 위해 각각의 화소의 합들을 총 화소의 값으로 나눈 뒤 10을 곱하여 각 화소의 분포의 총 합이 10이 되도록 하여 준다. 각각의 히스토그램 추출 방법은 <식>과 같이 수평, 수직의 히스토그램을 구한다.

$$H32_I = \frac{\sum_{i=1}^{32} I_{vij}}{\sum_{i=1}^{32}\sum_{j=1}^{32} I_{vij}} \quad (1)$$

$$V32_I = \frac{\sum_{j=1}^{32} I_{vij}}{\sum_{i=1}^{32}\sum_{j=1}^{32} I_{vij}} \quad (2)$$

그다음 구하여진 픽셀의 분포들을 왼쪽, 오른쪽, 가운데 가중치를 적용하여 세그룹으로 묶어 6개의 특징데이터를 구한다. 이 특징 데이터들을 기존의 자소 데이터들과의 유클리디안 거리를 계산하여 입력된 개체와 가장 가까운 값을 찾아내어 문자를 인식한다.

$$Ucdis = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (3)$$

2.2.1 역전파 알고리즘을 이용한 data 학습

본 논문에서는 출력된 결과값들을 이용하여 역전파 알고리즘 [3]을 통해 반복 학습하여 가중치와 데이터 값들을 최적화 하였다. 역전파 학습은 다층 퍼셉트론의 전개에 따른 가중치 및 임계값들에 관한 해를 반복적으로 구하는 일반적인 방법이다. 작은 학습율이 사용되는 경우에 아주 안정적인 최속 강하법에 속한다. 역전파 알고리즘은 다음과 같은 오차 제곱(목표값-출력값의 제곱)들을 합하여 얻어지는 목적함수를 최소화 시키는 방식으로 가중치와 임계값을 계산한다.

$$\varepsilon^2 = [t_q - f_{qk}]^2 \quad (4)$$

역전파 알고리즘의 단계를 간단히 정리하면 아래와 같다.

1. 학습데이터를 입력 노드에 적용하고 입력에 따른 출력을 계산한다.
2. 입력에 따른 출력과 원하는 출력간의 오차를 계산한다.
3. 오차에 따른 가중치의 증감여부를 결정한다.
4. 3단계에서 결정된 값으로 가중치를 갱신한다.
5. 모든 학습 데이터에 대해 오차가 적정 수준으로 감소하기 까지 1단계에서 4단계를 반복 학습한다.

단층이고 출력함수가 일차인 함수인 경우, 신경망의 가중치 갱신은 아래의 식으로 구성된다. 오차ε와 학습률μ, 그리고 입력값에 비례하여 가중치값을 갱신한다.

$$w(t+1)_i = w(t)_i - \mu\varepsilon_k x_{k+i} \quad (5)$$

3. 결 론

본 논문에서는 4방향 윤곽선 추적 알고리즘을 이용하여 자소를 분리한 뒤 K-means알고리즘을 통해 각각의 자음과 모음들의 픽셀정보와 Edge정보를 이용하여 한글 문자를 인식하는 방법을 제안하였다

제안된 알고리즘은 Intel Core 2 Duo 2.80GHz PC상에서 MATLAB R2007a을 이용하여 구현하였으며, font는 돋움, 굴림체, 바탕체, 고딕, 신명조, 궁서 휴먼매직체, HY바다, HY엽서, HY나무, 안상수가는체 그래픽 등 가장 많이 쓰는 12가지 폰트에 대해 임의의 크기의 100개의 인쇄체 문자를 이용하여 실험을 실행하였으며 인식 결과는 <표 2>와 같다.

문자	인식	오인식	인식률	문자	인식	오인식	인식률
ㄱ	8	1	89%	ㅏ	14	2	88
ㄴ	12	1	92%	ㅑ	12	0	100%
ㄷ	11	0	100%	ㅓ	8	0	100%
ㄹ	15	3	83%	ㅕ	6	0	100%
ㅁ	7	2	78%	ㅗ	5	1	83%
ㅂ	6	1	86%	ㅛ	4	0	100%
ㅅ	18	1	85%	ㅜ	12	2	86%
ㅇ	7	1	88%	ㅠ	5	0	100%
ㅈ	9	3	75%	ㅡ	5	1	83%
ㅊ	6	1	88%	ㅣ	11	2	84%
ㅋ	7	1	78%	ㅐ	5	0	100%
ㅌ	11	2	85%	ㅒ	7	1	88%
ㅍ	6	2	75%				
ㅎ	11	2	85%				
전체 인식률 : 88%							

표 2. 인식율 결과

임의의 100개의 인쇄체 문자를 이용하여 실험을 실행한 결과 8개의 데이터만을 가지고 처리하므로 인식의 처리속도도 매우 빨랐으며 다양한 폰트에 대해서도 인식률이 높음을 볼 수 있었다. 인식률의 저하 요인으로는 첫 번째로 다양한 폰트로 실험을 한 결과 서로 다른 폰트에서 같은 개체의 픽셀 분포가 크게 달라 잘못 인식되는 경우가 있었고, 두 번째로 좌, 우, 중앙의 픽셀 분포의 가중치에 따라 다른 결과를 보이기도 했다. 이러한 문제점을 개선하기 위해서는 반복적인 실험을 통해 가중치를 최적화하고, 세선화를 통해 획들의 굵기를 일정하게 변화시켜 준다면 더 높은 인식률을 얻을 수 있을 것으로 예상된다. 또한 한가지의 문자체에 대해서는 100%에 가까운 정확도를 얻을 수 있을 것으로 생각되며 더 나아가서는 필기체에 대한 인식도 가능할 것으로 보여진다.

[참 고 문 헌]

[1] 김성영·권태균·김민환, “추적에 의한 단순화된 윤곽선 추출”, 춘계학술발표논문집, Vol.2 No.1, pp.356-361, 1999년

[2] 허명회·손은진, “퍼지 K-평균 군집화의 재현성 평가”, 應用統計 (THE APPLIED STATISTICS), Vol.18, pp.1-11, 2003년.

[3] 장명숙·박기현, “역전파 알고리즘을 위한 개선된 훈련 방법”, 産業技術硏究所論文報告集 (BULLETIN OF THE INSTITUTE FOR INDUSTRIAL SCIENCE), Vol.16 No.1, pp175-182, 1993년