한국지능정보시스템학회논문지 제10권 제2호 2004년 11월(pp.65~77)

오류 역전파 알고리즘을 이용한 영문자의 폰트 분류 방법에 관한 연구

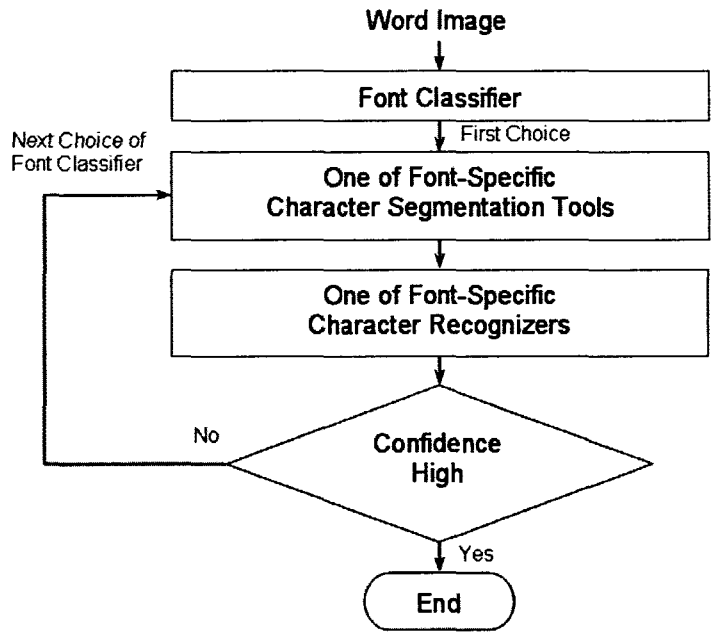
정민철
상명대학교 공과대학 컴퓨터시스템공학과
(mjung@smu.ac.kr)

본 연구에서는 영문 단어로부터 폰트를 분류하기 위해 연역적이고 국부적인 폰트 분류 방법을 제안한다. 이는 문자 인식 전에 한 단어에서 폰트를 분류하는 것을 말한다. 폰트 분류를 위해 활자 특성인 어센더(ascender), 디센더(descender)와 세리프(serif)가 사용된다. 입력 단어로부터 어센더(ascender), 디센더(descender)와 세리프(serif)가 추출되어 경사도 특징 벡터가 추출되고, 그 특징 벡터는 인공 신경망에 의해 입력 단어에 대한 2가지 폰트 스타일, 3가지 폰트 그룹, 7가지 폰트 이름이 분류된다. 제안된 연역적이고 국부적인 폰트 분류 방법은 폰트 정보가 문자 분할기와 문자 인식기에 사용될 수 있게 한다. 나아가, 특정 폰트에 따른 Mono-Font 문자 분할기와 Mono-Font 문자 인식기로 구성되는 OCR 시스템을 구성할 수 있는 것을 가능하게 한다. 실험 결과는 평균 95.4 퍼센트의 높은 폰트 분류율을 보였다. 본 논문에서 7가지 폰트분류를 위해 제안된 방법은 그 외 다른 폰류 분류에도 적용될 수 있다.

논문접수일 : 2004년 5월 게재확정일 : 2004년 11월 교신저자 : 정민철

1. 서론

영문으로 인쇄된 문서의 문자인식 분야에 있어서, OCR (Optical Character Recognition) 시스템에 알려진 폰트로 쓰여진 영문자를 인식하는 인식률은 높게 나타나다. 그 이유는 OCR 시스템이 알려진 폰트 내에서 공통된 규칙성을 발견하여 학습하기 때문이다. OCR 시스템에 알려지지 않은 폰트로 쓰여진 문자인식은 폰트의 다양성 때문에 인식률이 전자에 비해 급속히 떨어진다. 다양한 폰트로 쓰여진 문서의 문자인식에서 높은 인식률을 계속 유지하는 OCR 시스템을 만드는 것은 아직 풀어야 할 연구 과제이다. 이 문제를 해결하기 위해 Omni-Font OCR 시스템이 소개되었는데, 이 시스템은 한 문자에 대해 폰트의 분류 없이 일반적인 특징 벡터를 추출하여 문자 인식을 한다[1]. 그러나 이 논문에서 제시하는 폰트의 분류는 분류된 특정 폰트 내에 속한 문자 구조와 활자 특성에 대한 정보를 얻을 수 있다. 이러한 폰트 정보는 분류된 특정 폰트에 따른 문자인식을 하는 OCR 시스템을 구성할 수 있게 한다. 더 나아가 분류된 특정 폰트에 따른 문자 분할을 가능하게 한다. 즉, 폰트를 분류하면 특정 폰트에 따른 Mono-Font 문자 분할기와 Mono-Font 문자 인식기로 구성되는 OCR 시스템을 구성할 수 있다. [그림 1]은 이 논문에서 제안된 폰트 분류기의 역



[그림 1] 연역적이고 국부적인 폰트 분류(*a priori* and the local font classification)

할과 여러 개의 Mono-Font 문자분할 모듈기와 Mono-Font 문자 인식기로 구성되는 OCR 시스템을 나타낸다. 이 논문에서는 폰트 분류법을 제시한다.

2. 폰트 분류의 접근법

이 연구에서 폰트분류를 위해 사용한 접근법은 연역적이고 국부적인 접근 방식(*a priori* and the local approach)이다. 폰트 분류기에 위치에 따라 폰트 분류는 연역적인 방법과 귀납적인 방법으로 구분된다. 이 연구에서 제안된 연역적인 접근법은 문자 인식 전에 문자의 폰트를 알아내는 접근법이다. 따라서 폰트 분류기는 문자 인식전 전처리 단계(preprocessing step)에 위치하여, 폰트 정보가 문자 분할기와 문자 인식기에 사용될 수 있다. 귀납적인 접근법(*a posteriori* approach)은 문자 인식 후에 문자 인식기의 결과를 이용하여 폰트를 분류한다[2,3]. 폰트 정보는 문자인식 후, 문서를 원래의 폰트로 표현할 때 사용되어진다. 또한 폰트 분류기의 입력 텍스트의 길이에 따라 폰트 분류는 국부적인 방법(the local approach)과 전체적인 방법(the global approach)으로 구분된다. 이 연구에서 제안된 국부적인 접근법은 폰트 분류기의 입력으로 한 단어(one word)나 한 문자(one character)를 사용한다. 물론 스캔된 상태의 질이 떨어지는 문서에서는 한 단어나 한 문자의 입력으로 정확한 폰트를 분류하는 것은 심지어 인간에게도 불가능하다. 그러나 이 연구에서 제안된 폰트 분류법은 그러한 단어와 문자로부터도 최소 세리프(serif) 폰트와 산세리프(sans serif) 폰트 그룹

으로 분류하는 것이 가능하다. 전체적인 접근법은 한 페이지나 한 구문의 문서 전반에 걸쳐 주로 사용된 폰트를 분류하는 접근 방법이다. 참고문헌 [4]에서 S. Khoubyari는 한 문서에서 쓰여진 폰트를 분류하기 위해 영문 문서에서 사용 빈도수가 높은 단어들인 'the','of','and','a','to'등을 사용하여 문서 전체에 사용되어진 폰트를 분류하였다. 참고문헌[5]에서는 문서의 키워드 선택에 대해 학습을 통한 신경망의 접근 방법을 제안하였으며 단어의 문서 내 빈도와 역문서 빈도뿐만 아니라 단어의 위치도 또한 고려하였고 키워드 선택에 대하여 수식 모델에 비해 신경망의 접근이 정확도를 향상시키는 것을 제시하였다. 귀납적 폰트 분류와 전체적인 폰트 분류는 OCR 시스템의 결과를 이용한 방법으로 OCR 시스템의 인식률을 높이는 데 이용하기가 용이하지 않다. 그러나 본 연구에서 제안된 연역적이고 국부적인 폰트 분류는 OCR 시스템의 전처리 단계에서 폰트를 분류하여, 분류된 폰트 정보를 OCR 시스템이 문자 분할과 문자인식을 수행할 때 이용 가능하게 하여 OCR 시스템의 인식률을 높인다. 더 나아가 [그림 1]에서와 같이 특정 폰트의 문자에 강한 문자 분할기와 문자 인식기를 병렬로 구성하는 OCR 시스템을 구성할 수 있다. 연역적이고 국부적인 폰트 분류법을 사용하는 본 연구에서는 한 문자를 입력으로 받아, 두개의 폰트 스타일(upright와 slant), 세개의 폰트 그룹(serif, sans serif와 typewriter), 일곱개의 포스트스크립트 폰트 (Avant Garde, Helvetica, Bookman, New Century School Book, Palatino, Times와 Courier)를 문자 분할과 문자 인식 전에 분류한다. 위의 포스트스크립트 폰트는 레이저 프린터에 널리 표준으로 쓰이는 폰트이다. 폰트 분류기가 사용되는 응용 분야에 따라 폰트는 삭제 또는 첨가해 나갈 수 있다. 또한 폰트 분류기에 알려지지 않은 새로운 폰트는 위 폰트에 가장 가까운 폰트로 분류된다. 이 연구에서는 폰트 분류의 실험을 위해 경사도 특징 추출법과 인공 신경망이 이용되었다.

3. 폰트 분류의 방법

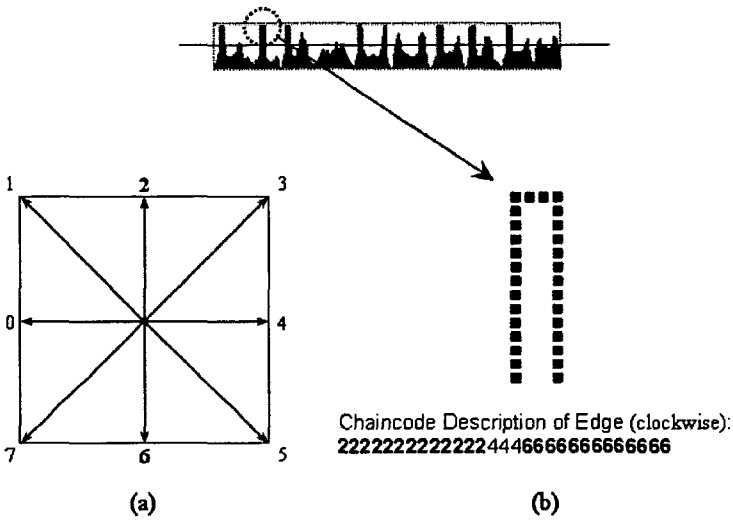
3.1 수직 투영 윤곽 분석법에 의한 폰트 스타일 분류

OCR 시스템에 입력된 문자가 upright(혹은 normal)스타일인지 slant(혹은 italic)스타일인지 분류하는 것은 중요하다. 왜냐하면, 먼저 문자를 분할할 때, 폰트 스타일에 따라 문자 분할 방식이 달라야한다. 예를 들면 upright스타일로 쓰여진 접합 문자는 수직으로 문자 분할이 가능하나, slant스타일로 쓰여진 접합 문자는 수직 문자 분할이 불가능하다. 또한 문자를 인식할 때, 문자의 내부 구조가 폰트 스타일에 따라 다르다는 것을 주시해야한다. 예를 들면 upright로 쓰여진 'm'과 slant로 쓰여진 '*m*'은 문자의 구조(또는 모양)가 다르다. 폰트 스타일을 분류하기 위해 [그림 2]와 같이 수직 투영 윤곽 분석법(the vertical projection profile)을 사용하였다.

[그림 2]에서 볼 수 있는 것처럼 upright스타일과 slant스타일의 수직 투영 윤곽은 완전히 다른 모양을 하고 있다. upright스타일로 쓰여진 단어의 수직 투영 윤곽에서는 직사각형 봉우리를 발견할 수 있고, slant스타일로 쓰여진 단어의 수직 투영 윤곽에서는 삼각형 봉우리를 주로 발견할 수 있다. 이 두 스타일의 수직 투영 윤곽을 분류하기 위해 [그림 3]에서처럼 체인코드가 사용되었다.

(a) Upright (b) Slant

[그림 2] Upright와 Slant스타일의 수직 투영 윤곽 분석



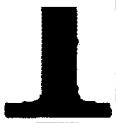
[그림 3] 직사각형 봉우리의 체인코드 분석

[그림 3]은 upright스타일의 수직 투영 윤곽 이미지를 상단 부와 하단 부로 나눈 후 상단 부에서 얻을 수 있는 직사각형의 체인 코드를 나타낸다. 직사각형의 체인코드는 연속적인 수직 방향 코드로 구성되는 반면, 삼각형의 체인 코드는 연속적인 수직 방향 코드(연속적인 2나 연속적인 6)가 구성되지 않는다.

3.2 세리프(serif)에 의한 폰트 그룹 분류

영문자에 있어 세리프(serif)는 문자의 주획의 위아래 양끝에 붙어 문자를 꾸미는 작은 획이다. 산세리프(sans serif)는 세리프(serif)가 없다는 뜻이다. 세리프(serif)는 폰트를 분류하는 데 가장 큰 특징 중 하나이다. 예를 들면 세리프(serif) 폰트 그룹에 속하는 Times 폰트의 'I', 산세리프(sans serif) 폰트 그룹에 속하는 Helvetica 폰트의 'I', typewriter 폰트 그룹에 속하는 Courier 폰트의 'I' 는 그 모양에 있어 크게 다르다. [그림 4]는 포스트스크립트 폰트의 문자 하단 부에서 추출한 다양한 세리프(serif) 모양을 나타낸다.

세리프(serif)의 존재 여부에 따라 세리프(serif)

Serif Font				Sans-Serif Font		Type-writer
Bookman	N.Century Schlbk	Palatino	Times	Avant Garde	Helvetica	Courier
						

[그림 4] 포스트스크립트 폰트의 문자 하단 부에서 추출한 다양한 세리프(serif)

폰트 그룹과 산세리프(sans serif) 폰트 그룹으로 나눌 수 있는데 세리프(serif) 폰트 그룹에는 Bookman, New Century School Book, Palatino와 Times 폰트가 있고, 산세리프(sans serif) 폰트 그룹에는 Avant Garde와 Helvetica 폰트가 있다. Courier 폰트의 문자도 세리프(serif)를 가지나 그림 4에서 볼 수 있듯이 세리프(serif)가 주획의 굵기와 같으며 크기 또한 상대적으로 제일 크다. Courier 폰트는 원래 타자기의 폰트로서 각 문자의 실제 폭에 관계없이 세리프(serif)를 이용하여 모든 문자의 폭을 같게 만든 것이 특징이다. 예를 들면 Courier 폰트로 쓰여진'I'와'W'는 여백을 점유하고 있는 폭이 동일하다. 즉, 같은 폭을 점유하기 위해 실제 폭이 훨씬 적은'I'의 세리프(serif)를 옆으로 길게 강조했다. 이와 같이 각 문자의 실제 폭과 관계없이 일정한 문자 폭을 가지는 typewriter 폰트는 fixed-pitch라고 하는데 접합문자가 드물며 문자 접합 시에도 문자 분할은 일정한 비에 따라 기계적으로 수행 가능하다. 그러나 컴퓨터의 등장에 따른 variable-pitch로 쓰여진 접합문자의 경우 문자 분할은 문자 구조의 선지식을 요구한다. 따라서 본 연구에서는 폰트를 3가지 그룹, 세리프(serif) 폰트 그룹, 산세리프(sans serif) 폰트 그룹과 타이프라이터(typewriter) 폰트 그룹으로 분류하였다.

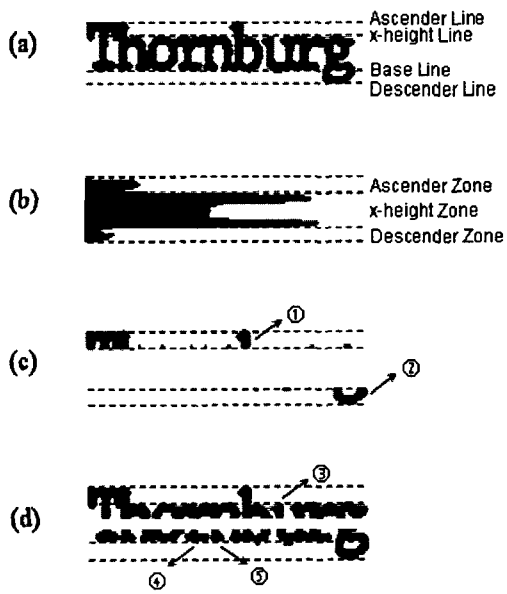
3.3 세리프, 어센더와 디센더를 이용한 폰트 분류

영문 알파벳의 문자 구조적 특징 중 하나는 영문 문자들이 어센더(ascender)나 디센더(descender)를 가진다는 것이다. 이러한 어센더(ascender)와 디센더(descender)는 [그림 5](b)에서 볼 수 있는 것과 같이, 단어의 수평 투영 윤곽 분석을 하면 3가지 영역, 즉 어센더(ascender)영역, x-height 영역과 디센더(descender) 영역으로 나눌 수 있다. 이 세 가지 영역은 어센더(ascender)라인, x-height라인, base라인과 디센더(descender)라인으로 구분된다. x-height영역은 모든 영문 단어가 점유하나 어센더(ascender) 영역과 디센더(descender) 영역은 영문 단어에 따라 점유 될 수도 있고 안 될 수도 있다. 이러한 영역의 분석 방법을 통해 각 문자의 세리프(serif), 어센더(ascender)와 디센더(descender)의 활자 특성을 추출하는 데, 추출하는 방법이 소문자로 구성된 단어와 대문자로 구성된 단어에 따라 다음과 같이 구분된다.

3.3.1 소문자의 폰트 분류

소문자로 구성된 단어는'어센더(ascender)영역과 x-height영역'또는'x-height영역과 디센더(descender) 영역'의 두 가지 영역만을 가지거나 세 가지 영역 모두를 가질 수 있다. [그림 5]는 세 가지 영역 모두를 가지는 단어의 예를 보였다. [그림 5](a)에서처럼 처음 문자가 대문자인 단어(Capitalized word)도 이 구분에 속한다.

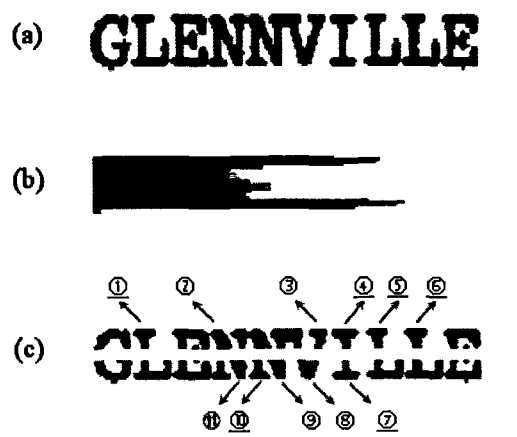
[그림 5](c)는 x-height 영역을 제거한 후 남은 이미지이다. 이 이미지로부터 세리프(serif)인 ①과 디센더(descender)인 ②를 얻을 수 있다. 'T'상단부분은 어센더(ascender)의 후보로 검증되나 그 폭이 어센더(ascender)가 되기에는 너무 커서 제외된다. 세리프(serif)를 추출하기 위해, [그림 5](d)에서와 같이 x-height의 중앙부분을 제거하고 남은 이미지를 분석한다. 그 이미지로부터 세리프(serif)인 ③, ④와 ⑤를 얻을 수 있다. 나머지들은 세리프(serif)가 되기에는 폭이나 높이가 문자 크기와 비교할 때 너무 커서 모두 제외된다. 접합 문자에서 발생하는 접합된 세리프(serif)는 이 추출 방법에서 모두 제외된다.



[그림 5] 소문자로 구성된 단어의 활자 특성 추출: (a) Capitalized word, (b) 입력 단어의 수평 투영 윤곽 분석, (c) x-height 영역을 제거한 후 남은 이미지, (d) x-height의 중앙부분을 제거하고 남은 이미지

3.3.2 대문자의 폰트 분류

모두 대문자로만 구성된 단어는 [그림 6](b)의 수평 투영 윤곽 분석에서 볼 수 있는 것과 같이 한 영역만을 가진다. 따라서 단어의 중앙부분을 제거하고 남은 이미지를 분석하면 세리프(serif)를 얻을 수 있다. [그림 6](c)에서 ①, ④, ⑤, ⑦과 ⑩은 세리프(serif)이다. 본 연구에서 세리프(serif)는 모양에서 대칭성을 가진 것으로 한정하였다. ①과 ⑪은 모양에서 비대칭이므로 세리프(serif)에서 제외된다. 나머지들은 모두 세리프(serif)가 되기에는 문자의 크기와 비교 할 때 모두 너무 폭이 넓어서 제외된다.



[그림 6] 대문자로만 구성된 단어의 활자 특성 추출: (a) All capitals, (b) 입력 단어의 수평 투영 윤곽 분석, (c) 입력 단어의 중앙부분을 제거하고 남은 이미지

3.4 경사도 특징 추출

경사도 특징 추출(gradient feature extraction)은 필기체 인식을 위해 개발되었다[6]. 이 연구에서는 추출된 활자 특성인 어센더(ascender), 디센더(descender)와 세리프(serif)들로부터 경사도 특징 추출이 수행된다. 경사도는 Sobel 연산자에 의해 수행되는데 [그림 7]은 Sobel 연산자 템플릿을 나타낸다.

-1	0	1
-2	0	2
-1	0	1

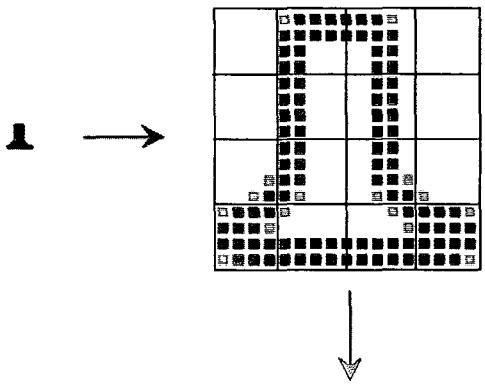
Δx

1	2	1
0	0	0
-1	-2	-1

Δy

[그림 7] Sobel 연산자 템플릿:
참고문헌[7]에서 인용

추출된 활자 특성 이미지의 각 픽셀은 Sobel 연산자 템플릿과 Convolution되어 Δx 성분과 Δy 성분 값이 중앙 픽셀로 그 값이 저장되어진다. 즉, 중앙 픽셀의 경사도는 주변 8 픽셀의 함수로서 계산되어진다. 경사도는 0에서 2π라디안의 범위를 가지며 계산의 단순화와 속도를 감안해 8개 영역으로 양자화하고 정반대 방향은 같은 것으로 간주하여 4개의 경사도를 나타내면 경사도는 0, $\pi/4$, $\pi/2$ 와 $3\pi/4$로 나타낼 수 있다. 추출된 활자 특성 이미지는 4× 4 grid로 영역을 나누고, 각 영역에 있는 픽셀의 경사도를 히스토그램으로 누적 수치화 하여 지정된 임계값을 초과하는 경사도를 카운트한다. 이러한 방법은 4× 4× 4, 즉 64개의 특징 벡터를 구성한다. 이 특징 벡터는 인공 신경망의 입력 벡터가 된다. [그림 8]은 추출된 세리프(serif)의 경사도 맵과 그 특징 벡터를 나타낸다.



0.00 0.00 0.00 0.00	0.50 0.00 0.17 0.00	0.33 0.00 0.00 0.00	0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00	0.00 0.00 1.00 0.00	0.00 0.00 1.00 0.00	0.00 0.00 0.00 0.00
0.17 0.00 0.17 0.00	0.00 0.50 0.33 0.00	0.17 0.00 0.50 0.50	0.17 0.00 0.00 0.00
0.33 0.00 0.00 0.00	1.00 0.00 0.00 0.00	1.00 0.00 0.00 0.00	0.00 0.00 0.17 0.00

[그림 8] 추출된 세리프(serif)의 경사도 맵과 그 특징 벡터

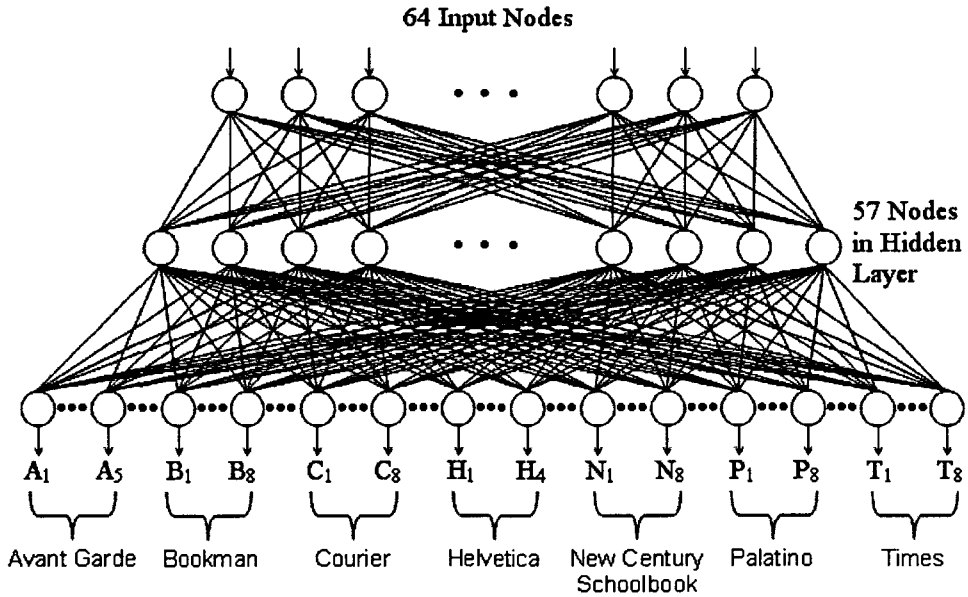
4. 인공 신경망 Classifier

참고 논문[7]에서는 인쇄된 한글을 인식하기 위해 인식단위 결합을 한 후 다중 신경망을 이용하였다. 본 논문에서는 영문자 폰트 분류를 위해 오류 역전파 알고리즘을 이용한 인경 신경망 classifier가 이용되었다. 즉 폰트 분류기는 64개의 입력 노드를 가지고, 57개 노드의 hidden layer를 가지며, 49개의 출력노드를 가지는 3-layer back propagation 인공 신경망으로 구성된다. [그림 9]는 그 구성을 나타낸다.

hidden layer의 노드 개수는 입력노드와 출력노드수의 평균값으로 초기에 주어진 후, 트레이닝과 테스팅의 절차를 거쳐 가장 최적화 된 값인 57을 선택하였다. 입력 단어로부터 모든 어센더(ascender), 디센더(descender)와 세리프(serif)가 처리된 후 폰트 분류기는 그 단어에 대한 폰트를 분류한다. 예를 들어 입력 단어로부터 한 개의 어센더(ascender), 한 개의 디센더(descender)와 세 개의 세리프(serif)를 추출했다면 추출된 5개의 활자 특성이미지는 5개의 특징 벡터 세트로 바뀌어 인공 신경망에 입력되며, 49개의 출력 노드는 누적된 confidence값을 가진다. 각각의 폰트에 대한 누적된 confidence값은 Times 폰트의 경우 다음과 같이 계산되어진다.

$$Times = \sum_{i=1}^{8}\left(\sum_{j=1}^{n} C_j(T_i)\right) \qquad (1)$$

위 식에서 n은 추출된 활자 특성이미지 개수이며, $C_j(T_i)$는 Times 폰트의 각 confidence값이며, 8은 Times 폰트의 전체 활자 특성이미지 개수이



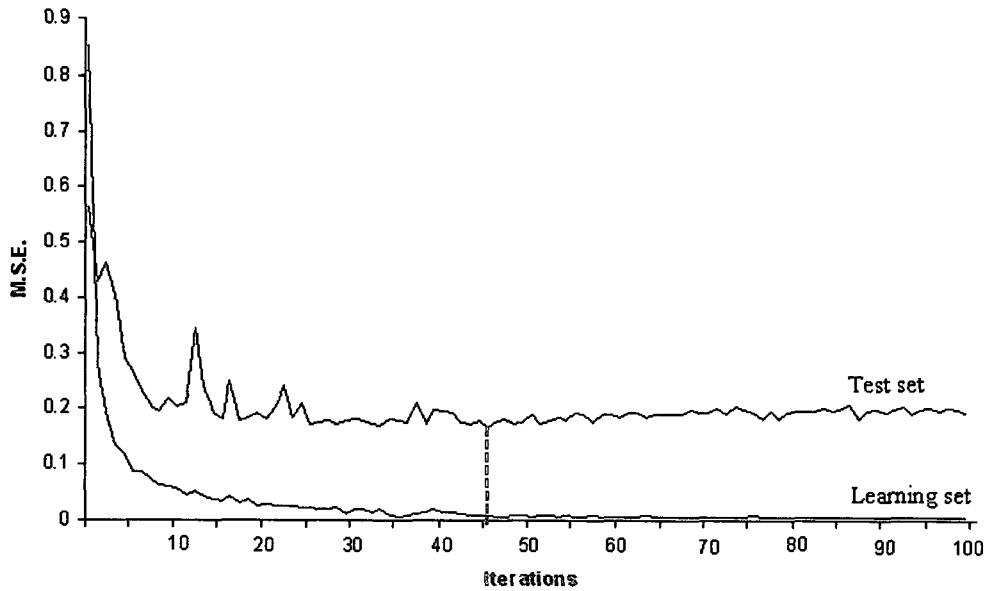
[그림 9] 3-Layer Back propagation 인공 신경망의 구성도

다(2 ascenders, 4 descenders와 2 serifs). 각 누적된 confidence 값 중 최대값을 가지는 폰트가 입력 단어의 폰트로 분류된다.

5. 실험과 결과

실험에 사용된 이미지는 레이저 프린터로 인쇄되어, 300dpi로 스캔되고, 같은 파라미터를 가지고 이미지 이진화가 되었다. 트레이닝 데이터는 2120개의 이미지인데 활자 크기는 10-, 12- 와 14-point size가 혼합되었다. 테스팅 데이터는 1060개의 이미지인데 역시 0-, 12- 와 14-point size가 혼합되었다. 각 이미지는 폰트 사이즈에 관계없이 일정한 크기를 유지하게 하기 위해 크기 정규화를 하였다. 인공 신경망의 트레이닝은 learning rate = 0.1과 momentum = 0.5로 하여 수행되었다. 트레이닝 과정은 MSE (Mean Squared Error)에 의해 모니터 되었으며, 트레이닝 반복학습(iteration)의 함수로서 MSE 의 값을 [그림 10]에 나타내었다. MSE 값은 54개의 hidden 노드에서 가장 적은 값을 나타냈으며 0.0069에 수렴하였다.

트레이닝 결과는 테스팅 데이터에 의해 평가되어진다. 테스팅 과정 또한 MSE 에 의해 모니터 되어졌다. 테스팅 반복학습(iteration)의 함수로서 MSE 의 값을 그림 10에 나타내었다. 테스팅 과정에서 인공 신경망은 46번의 반복학습에서 MSE 의 국부 최소값(local minima) 0.1648을 가진다. 따라서 46번 반복 학습한 트레이닝의 weight matrix가 인공 신경망의 최대 일반화를 하는 것으로 선택되어진다. 이 값 이상의 트레이닝은 지나치게 학습(over- trained)된 것이다. <표 1>은 선택된 폰트에 대한 폰트 분류 결과를 나타낸다. 폰트 분류는 트레이닝 이미지와 테스팅 이미지와 다른 1000개의 단어 이미지로 수행되었다. 그 결



[그림 10] 인공 신경망의 Learning Curve와 Test Curve

<표 1> 폰트분류기의 평균 분류율

(단위: 퍼센트)

Point size		10 pts	12 pts	14 pts
Serif font	Bookman	91.2	96.7	96.8
	New Century Schoolbook	93.5	95.0	97.8
	Palatino	91.1	94.3	97.2
	Times	93.3	95.9	97.3
Sans-serif font	Avant Garde	94.6	97.4	98.5
	Helvetica	94.1	96.5	97.2
Typewriter	*Courier*	94.6	95.4	97.0

과 평균 95.4 퍼센트의 높은 폰트 분류율을 보였다. 10-point size로 쓰여진 단어의 폰트 분류율은 다소 낮은데, 그 이유는 작은 폰트로 쓰여진 문자의 세리프(serif) 또한 그 모양이 너무 작아 불분명하고 외부 노이즈에 민감하기 때문이다. 세리프(serif) 폰트들이 산세리프(sans serif) 폰트들보다 다소 낮은 분류율을 보이는 데, 이는 세리프(serif)가 더 복잡한 구조를 가지기 때문이다. 폰트 사이즈가 커짐에 따라 폰트 분류율도 증가됨을 볼 수 있다. 폰트 분류의 에러는 주로 같은 폰트 그룹 내에서 발생된다. 그러한 에러는 문자 분할기와 문자 인식기에게 있어 종종 받아들일 수 있는 에러이다. 왜냐하면 같은 폰트 그룹 내에 있는 문자의 경우 그 문자의 구조가 대동소이 한 경우가 많기 때문이다.

6. 결론

본 연구에서는 영문 단어로부터 폰트를 분류하기 위해 연역적이고 국부적인 폰트 분류 방법을 제안했다. 폰트 분류를 위해 활자 특성인 어센더(ascender), 디센더(descender)와 세리프(serif)가 사용되었다. 따라서 폰트 분류의 정확도는 어센더(ascender), 디센더(descender)와 세리프(serif)에 의존한다. 어센더(ascender), 디센더(descender)와 세리프(serif) 등의 활자 특성은 문서가 낮은 해상도를 가지고 스캔되거나 스캔된 이미지가 degraded 되었을 때는 소실됨으로, 본 연구에서 제안된 연역적이고 국부적인 폰트 분류 방법은 한계를 가진다. 그러나 적절한 해상도(200dpi 이상)를 가지고 스캔된 문서에 대해서는 실험의 결과에서 나타내듯이 높은 폰트 분류율을 보인다. 이 폰트 분류의 결과는 OCR 시스템의 문자 분할 모듈이나 문자 인식 모듈에서 이용되어 OCR 시스템의 전체 인식률을 높이는 데 이용 될 수 있다.

Acknowledgement

본 연구는 상명대학교 교내 연구비 지원으로 수행되었음.

참고문헌

[1] Kahan S. and T. Pavlidis and H.S. Baird, "On the recognition of printed characters of any font and size", *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol.9, No.2, pp. 274-288, 1987.

[2] Shi H. and T. Pavlidis, "Font Recognition and Contextual Processing for more accurate text recognition", *4th International Conference on Document Analysis and Recognition*, pp. 39-41,1997

[3] Zramdini A, "Study of Optical Font Recognition based on Global Typographical Features", *Ph.D. Dissertation*, University of Fribourg, Switzerland, 1995.

[4] Khoubyari S and J.J. Hull, "Font identification using visual global context", *SPIE Vol. 2181 Document Recognition*, pp. 116-124, 1994.

[5] 조태호, "문서의 키워드 추출에 대한 신경망 접근", *한국정보과학회 학술발표논문집*, 27권, 2호, pp. 317-319, 2000.

[6] Favata J. T., "A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition", *International Journal of Imaging Systems and Technology*, Vol.7, pp. 304-311, 1996.

[7] Srikantan G, S.W. Lam and S.N. Srihari, "Gradient-based Contour Encoding For Character Recognition", *Pattern Recognition*, Vol.29, No.7, pp. 1147-1160, 1996.

[8] 임길택, "다중 신경망을 이용한 인식단위 결합 기반의 인쇄체 문자인식", *한국정보처리학회*, 10권, 7호, pp. 777-784, 2003.

Abstract

Font Classification using Back Propagation Algorithm

Minchul Jung*

This paper presents *a priori* and the local font classification method. The font classification uses ascenders, descenders, and serifs extracted from a word image. The gradient features of those sub-images are extracted, and used as an input to a neural network classifier to produce font classification results. The font classification determines 2 font styles (upright or slant), 3 font groups (serif, sans-serif, or typewriter), and 7-font names (PostScript fonts such as Avant Garde, Helvetica, Bookman, New Century Schoolbook, Palatino, Times, and Courier). The proposed *a priori* and local font classification method allows an OCR system consisting of various font-specific character segmentation tools and various mono-font character recognizers. Experiments have shown font classification accuracies reach high performance levels of about 95.4 percent even with severely touching characters. The technique developed for the selected 7 fonts in this paper can be applied to any other fonts.

Key words : Font classification, Optical character recognition(OCR), Artificial neural network, Gradient feature vector, Chain code

* Department of Computer System Engineering, Sangmyung University