

무제약 필기체 숫자를 인식하기 위한 병렬 파이프라인 다중 인식기의 구조와 결합 방법

최용호*, 이호현, 조범준
조선대학교 컴퓨터공학부

Combinations Method and Parallel Pipeline Multiple Recognizer Structure for Recognizing Unconstrained Handwritten Numerals

YongHo Choi, HoHyun Lee, BeomJoon Cho
School of Computer Engineering, Chosun University.

E-mail : yhchoi@ai.chosun.ac.kr, newcom@ai.chosun.ac.kr, bjcho@mail.chosun.ac.kr

요 약

숫자를 인식하는 방법에는 여러 가지가 있지만 단일 인식기를 구성하는 경우보다 다중 인식기를 이용하는 방법이 뛰어나다는 연구 발표가 있었다. 그래서 다중 인식에 대한 연구가 활발히 진행되고 있는데, 다중 인식기를 이용하는 방법에는 크게 직렬 조합형과 병렬 조합형이 있는데, 직렬 조합형은 인식기를 파이프라인 처럼 구성하여 순차적으로 인식하는 방법이고, 병렬 조합형은 인식기를 병렬로 구성하여 인식기들의 결과를 조합하여 얻어내는 방법이다. 본 논문에서는 무제약 필기체 숫자를 인식하기 위한 병렬 파이프라인 다중 인식기의 구조와 결합 방법을 제안 하고자 한다. 조선대학교 필기체 숫자 데이터를 이용하여 실험한 결과 기존의 방법보다 비교적 높은 인식률을 나타내었다.

1. 서론

정보화 사회에서 방대한 양의 데이터를 보다 신속하고 정확하게 처리하기 위해서는 사람들의 궁극적인 생각일 것이다. 그 중에서 필기체 숫자 인식은 수표, 신용카드 매출표, 세금계산서, 각종 전표 및 문서 등은 작업의 자동화와 전자 문서화를 지향하여 검색, 수정 등이 매우 용이하며 업무 처리 속도 등을 향상 시키는 것은 중요한 부분일 것이다. 또한 이들은 모두 금액을 다루는 경우가 많기 때문에 높

은 수준의 인식률과 신뢰도를 필요로 한다. 그러므로 숫자인식에 관련된 연구의 핵심은 인식 성능의 극대화를 추구해야 한다. 그래서 과거에 수기로 작성된 문서, 혹은 컴퓨터를 이용하기 힘든 곳에서 작성되는 문서의 경우 전자 문서화 하기 위하여 재입력을 해야만 했다. 따라서 여기에 투입되는 시간이나 비용을 절감하기 위하여 문서인식에 관한 연구가 활발히 진행되고 있고, 인식률의 극대화가 가장 중요한 부분 일 것이다. 인식 성능을 향상 시키기 위한 연구는 주로 인식기의 개발에 초점을 두어왔고,

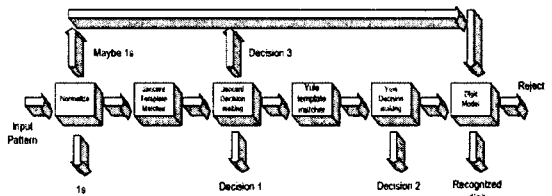
많은 인식기가 개발 되었지만 만족스럽지 못한 결과가 나왔다. 그래서 최근엔 단일 인식기를 사용하여 구성하는 시스템에 대한 연구가 발표되고 있다 [1,2,3]. 여기서 다중 인식 시스템은 인식기를 배치하는 방법에 따라 병렬 조합형 (Parallel Combination Type)과 직렬 조합형 (Serial Combination type) 두 가지로 나뉘게 된다. 병렬 조합형은 2개 이상의 인식기를 병렬로 구성하여 각각의 인식기에서 나오는 결과치를 조합하여 최종적인 인식 결과를 출력하게 되는데, 그러나 각각의 인식기에서 올바른 출력을 하더라도 조합기에서 오인식할 수 있는, 즉 조합기에 의존도가 높다는 단점이 있다. 직렬 조합기는 각각의 인식기를 파이프라인처럼 직렬로 연결하는 방식으로 쉬운 패턴은 앞에서 인식이되고 어려운 패턴은 여러 단계를 거치면서 인식된다. 하지만 인식기의 나열 방법이 시스템 구성원의 관건이기 때문에 시스템 구성방법에 다양성이 없고 개별 인식기의 성능에 비해서 결합 후의 성능이 그다지 향상 되지 않는다는 단점이 있다.

본 연구에서는 다수의 인식기를 구성하는 방법과 개별 노드에 결과값에 대한 조합 방법에 대하여 연구를 하였다. 이러한 연구는 다수의 인식결과를 결합하는 방법의 향상을 위하여 수행 되었으나 대부분이 인식기간의 의존관계를 고려하지 않았다. 대신, 다수 인식기가 독립이라고 가정하고 인식결과를 결합하였다. 만일 매우 높은 의존관계를 지닌 열등한 인식기가 기존의 다수 인식기 시스템에 추가되고, 이전의 결합 방법으로 다수의 인식 결과를 결합한다면 인식 성능이 저하되거나 결합된 최종 인식결과가 편향 될 수도 있다. 본 연구 동기는 이러한 기존 결합 방법의 단점을 극복하려는 것이다. 그래서 인식기의 성능을 최대한 이용하기 위해 기존의 직렬 조합이나 병렬 조합을 결합한 무제약 필기 숫자를 인식하기 위한 병렬 파이프라인 다중 인식기의 구조와 결합방법을 제안한다. 본 연구에서는 6개의 인식기와 3개의 검증기를 사용 하였으며, Voting방법을 이용하여 다수의 인식 결과를 결합하는 방법을 이용하였다.

2. 관련연구

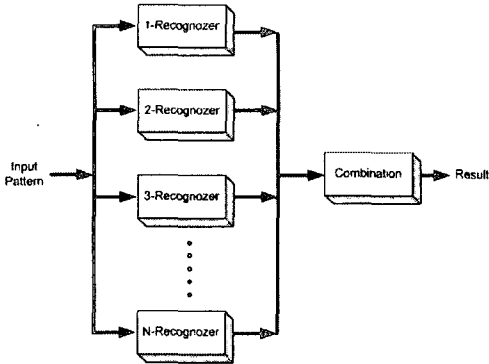
2.1 직렬 조합

복수개의 인식기를 직렬로 연결하는 메커니즘으로서 첫 번째 인식기에 의해 기각된 패턴들을 가지고 두 번째 인식기를 훈련시키는 방법이다. Garder의 인식기는 템플릿 기반 인식기와 모델 기반 인식기를 사용하였다. 인식 방법에 있어서도 그림과 같이 3가지의 결정 규칙을 이용하였다. 결정 규칙 1은 Jaccard top match의 값이 적당한 조건에 부합하면 여기에서 입력 패턴의 인식 절차는 끝이 나게된다. 결정 규칙 2는 Jaccard top match와 Yule top match의 값이 적당한 조건에 부합하면 여기에서 여기에서 입력 패턴의 인식 절차는 끝이 나게 된다. 결정 규칙 3은 결정 규칙 1의 조건에 부합하지 않을 때 패턴의 인식과정이 모델 매치로 이동된다[4].



2.2 병렬 조합

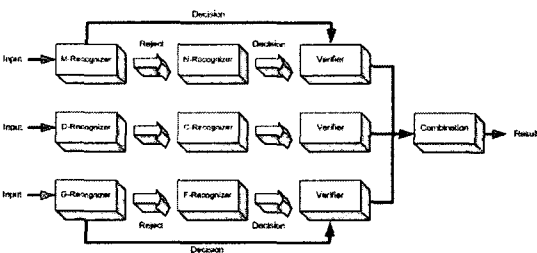
다수의 서로 다른 인식기를 동시에 사용하고, 그들의 인식 결과를 결합 함으로써 인식 성능을 향상 시키려는 다수 인식기 시스템에 대한 연구가 시작되었다. 다수 인식기 시스템은 다수 인식기의 집합체와 이들의 인식 결과를 결합하는 결합기로 구성되는데, 다수의 상호 보완적인 특징 점이나 인식 알고리즘이 동시에 사용된다면 인식 성능을 향상 시킬 수 있다는 가정에서 연구되었다. 인식 성능의 향상을 위하여 다수 인식기를 결합하는 연구 아이디어는 낙관적으로 보이나, 시스템의 성능이 결합기의 특성에 의존한다는 단점이 있다.



3. 병렬 파이프라인

병렬 파이프라인 시스템은 병렬 조합형과 직렬 조합형을 병합한 방법으로 병렬 조합형의 장점인 여러 개의 인식기 결과를 조합 할 수 있다는 점과, 직렬 조합의 장점이라 할 수 있는 여러 단계의 인식으로 인식률의 극대화를 시킬 수 있으므로 다중 노드에서의 높은 인식 결과값을 가지고 조합을 하게 되므로 오인식을 줄이고 인식 성능을 향상 시킬 수 있다.

제안하는 시스템에서는 서로 다른 특징 벡터값을 입력 값으로 사용하였고, 6개의 인식기와 3개의 검증기를 사용 하였으며 각각의 파이프라인 노드에 2개의 인식기와 1개의 검증기를 파이프라인 형태로 연결하여 그림 3.1과 같이 병렬 형태로 배치하여 3개의 결과값이 출력이 되도록 설계하였다.



3.1 1, 3단계 파이프라인

첫번 인식기에서는 어떤 입력 영상에 대해서 출력된 결과값을 파이프라인의 마지막 단계인 검증기

로 입력하여 결과값에 대해 검증을 하게 된다. 검증 값이 참이면 인식 결과대로 출력을 하게 되고 거짓이면 기각을 하게 된다. 두번째 인식기는 입력 패턴이 첫 번째 인식기에 의해서 거부되었을 때만 호출되고, 첫 번째 단계에서 거부된 패턴들을 이용하여 학습을 하였다. 아래에서는 이런 구조의 알고리즘 방법을 의사코드(Pseudo Code)로 표현 하였다.

```
string uf_main_01(string input_data)
{
    string output_data;
    string ls_result[2];

    ls_result[0] = uf_Recognizer_1(input_data);

    if ls_result[0] = 'Reject' then
    {
        ls_result[1] = uf_Recognizer_4(input_data);
        if ls_result[1] = 'ERROR' then
        {
            return 'Reject';
        }
        else
        {
            output_data = uf_Verifier_1(input_data,
                ls_result[1], 'Reject');
        }
    }
    else
    {
        output_data = uf_Verifier(input_data, ls_result[0],
            'Reject');
    }
    return output_data;
}
```

3.2 2단계 파이프라인

첫 번째 인식기에서 결정된 결과값을 출력 하였을 때 두 번째 인식기에서는 다시 한번 입력 영상에 대해 인식을 하게 된다. 첫 번째 인식기 결과 값과 두 번째 인식기 결과 값이 같을 경우 하나의 결과 값만이 검증기에서 검증을 하게 되고, 값이 틀릴 경우 2개의 결과 값을 검증 하게 된다. 아래에서는 이런 구조의 알고리즘 방법을 의사코드(Pseudo Code)로 표현 하였다.

```
string Uf_main_02(string input_data)
{
    string output_data;
    string ls_result[2];

    ls_result[0] = uf_Recognizer_2(input_data);
    ls_result[1] = uf_Recognizer_5(input_data, ls_result[0]);
    if ls_result[0] = ls_result[1] then
    {
        output_data = uf_Verifier_20(input_data, ls_result[1],
            'Reject');
    }
    else
    {
        output_data = uf_Verifier_21(input_data, ls_result[0],
            ls_result[1]);
    }
}
```

4. 인식기 모듈

각각의 인식기들은 다층 퍼셉트론 분류기로 이루어졌으며, 인식기에서 학습을 위하여 사용하는 신경망은 입력층, 1개의 은닉층, 출력층으로 구성되어 있다.

4.1 M-인식기

일반적인 숫자인식에서 가장 많이 사용하는 Mesh Feature(그물눈 특징)를 사용하였는데, 본 논문에서 숫자영상은 15 24의 크기로 정규화를 하였다. 각 부분 영역을 3 3의 크기를 가지는 윈도우로 사용할 때 숫자는 5 8의 영역으로 나누어진다.

4.2 N-인식기

교차 횟수 특징(Crossing Number Feature)은 숫자의 구조적 형태를 나타내는 특징으로서, 숫자의 수직 축과 수평 축으로 검출 선을 주사하였을 때 검출선과 검은 화소의 만나는 횟수로서 표현된다.

4.3 D 인식기

부분 영역에 대한 방향 벡터 특징(Directional vector feature)은 4가지 방향의 3 3 마스크를 숫자 영상에 적용했을 때 각 방향에 대한 화소의 누적 분포로써 표현된다[19]. 4가지의 마스크는 수직, 수평, 45 및 135 방향을 나타낸다. 방향 벡터 특징의 추출은 7 10의 크기로 정규화된 영상에 대해, 각 방향별 화소의 분포 형태를 구함으로써 수행된다.

4.4 G-인식기

숫자 영상의 모서리(edge)에 대한 기울기 방향 특징(gradient feature)은 Sobel 연산자로부터 구해진 문자의 방향 성분을 이용한다. Sobel 연산자는 1차 미분기로 동작함에 따라 영상에서 화소의 변화가 존재하는 전경(foreground)과 배경(background)의 경계 영역을 추출할 수 있다. 128-71-10의 구조를 가지는 다층 퍼셉트론 신경망 구조의 입력 벡터로 사용된다.

4.5 C-인식기

숫자의 윤곽선 방향 성분 특징은 인접한 화소사이의 관계를 이용한다. 이 특징의 추출과정은 먼저 숫자 영상의 윤곽선을 시계 방향으로 추적 하면서 현재 화소의 위치와 다음 화소의 위치와의 관계를 구한다. 인접한 화소와의 관계는 수평, 수직, 45 및 135 방향에 대하여 특정한 숫자로 대응시킨다. 예로서 수평 방향은 1, 135 방향은 2, 수직 방향은 3, 그리고 45 방향은 4로 대응시킬 수 있다.

4.6 F-인식기

교차 거리 특징(Cross Distance Feature)은 영상의 MBR(Minimum Bounding Rectangle)을 기준으로 설정하고, 이 기준 점으로부터 수직 및 수평 방향으로 검출선을 주사할 때 검은 화소를 처음 만날 때까지의 거리로서 나타낸다.

5. 검증기

본 논문에서 사용하는 검증기는 입력 영상의 구조적 특징을 탐지하고 이를 이용하여 검증한다. 구조적인 특징으로 사용된 방법은 교차 횟수 특징(Crossing Number Feature)과 교차 거리 특징(Cross Distance Feature)을 이용하여 검증을 하였다. 구조적인 특징을 탐지하여 인식기에서 결정된 결과값에 벗어나는 검증 결과 값이 나올 경우는 무조건 기각하고, 검증 결과값이 인식기의 결과값과 같을 경우만 결합기에 입력이되는 형태로 구현 하였다. 아래에서는 검증기 알고리즘을 의사코드(Pseudo Code)형태로 나타내고 있다.

```
string of_Verifier(string input_data, string first_result, string second_result)
{
    boolean lb_result[4];
    long ll_count[2];

    lb_result[0] = sub_Verifier0(input_data, first_result);
    lb_result[1] = sub_Verifier1(input_data, first_result);
    // ll_count[0] = True, False 개수
    lb_result[2] = sub_Verifier0(input_data, second_result);
    lb_result[3] = sub_Verifier1(input_data, second_result);
    // ll_count[1] = True, False 개수

    if lb_result[0] == "TRUE" then
        ll_count[0] ++;

    if lb_result[1] == "TRUE" then
        ll_count[0] ++;

    if lb_result[2] == "TRUE" then
```

```

        ll_count[1] ++;
    if lb_result[3] == "TRUE" then
        ll_count[1] ++;
    if ll_count[0] > ll_count[1] then
    {
        Return first_result;
    }
    else if ll_count[0] < ll_count[1] then
    {
        Return second_result;
    }
    else
    {
        if first_result = second_result then
        {
            Return first_result;
        }
        else
        {
            Return "Reject";
        }
    }
}
    
```

6. 결합기

본 논문에서는 다중 노드에서 출력되는 결과값을 조합하기 위해 Voting 방법을 사용하였다. 결합 방법 중의 한 방법인 Voting 방법은 다수의 결정에 따른다는 의미를 가지고 있는데, Voting 방법은 가장 흔한 결합 방법으로, 다수의 결정에 따른다는 의미를 가지고 있는데, 이 방법의 장점은 단순하고 별도의 기억 장소를 필요로 하지 않는다는 것이고, 단점으로는 개개의 방법이 행동의 성향과 자질에 상관 없이 같은 가중치를 가지고 취급 되어 진다는 사실이다. Voting 규칙을 설명하기에 앞서 표현의 간결을 위해 다음과 같은 이진 특성 함수를 도입 하기로 한다.

$$T_k(x \in C_i) = 1 \text{ when } e_k(x) = i \text{ and } i \in \Lambda$$

$$0 \text{ otherwise}$$

여기서 Λ 은 부류들의 집합을 나타내는데, 숫자의 경우 $\Lambda = \{0,1,\dots,9\}$ 가 된다. $e_k(x) = i$ 는 데이터 x 를 입력으로 받아 결과로써 i 라는 부류를 출력하는 인식 방법 k 를 나타내고 있는데, k 값은 $1 \leq k \leq \alpha$ (인식기 개수)의 값을 가진다.

7. 실험 결과 및 분석

본 실험 연구에서 사용되었던 필기체 숫자 데이터는 조선대학교 학생 500명에게 (가로)2.3cm (세

로)2.45cm 박스 10개의 안에 각각 0-9까지를 수성 사인펜으로 숫자의 굵기, 숫자의 크기, 기울기, 형태 등에 제안을 두지 않고 기입하도록 하여, 총 5000자의 데이터로 구성을 하였다. 그 중에서 훈련 데이터는 3000개의 숫자를 사용하였고, 테스트 데이터는 2000개의 숫자를 각각 사용 하였다. 학습 방법은 BP(Back Propagation) 알고리즘을 이용하여 임계치에 대한 조정을 하였고, 분류 과정에는 다층 퍼셉트론(MLP)을 이용하여 필기체 숫자 인식 실험을 하였다. 먼저 개별적인 파이프라인 인식기의 성능을 알아보기 위하여 실험을 하였고, 마지막으로 최종적인 병렬 파이프라인 인식 성능 실험을 하였다.

1단계 파이프라인 인식 결과는 표.1 에서와 같은 결과를 얻게 되었는데, 총 64개의 숫자를 오인식하여 3.2%의 오류율을 나타냈으며, 전체 2000개의 숫자 중 1827개의 숫자를 정인식을 하여 91.35%의 인식률을 나타내었는데 2와 8에서 잦은 오류를 발생하였다.

오류	64 / 2000	3.200(%)
기각	109 / 2000	5.450(%)
정인식	1827 / 2000	91.35(%)
신뢰도	1827 / 1891	96.61(%)

표.1 1단계 파이프라인 인식 결과

2단계 파이프라인 숫자별 인식 결과는 표.2 에서와 같은 결과를 얻게 되었는데, 총 61개의 숫자를 오인식하여 3.05%의 오류율과 1846개의 숫자를 정인식하여 92.30%의 인식률을 나타냈다. 2단계 노드의 결과 숫자 4와 0, 9에서 잦은 오인식을 하였다.

오류	61 / 2000	3.050(%)
기각	93 / 2000	4.650(%)
정인식	1846 / 2000	92.30(%)
신뢰도	1846 / 1907	96.80(%)

표.2 2단계 파이프라인 인식 결과

3단계 파이프라인에서 숫자별 인식 결과는 표.3에서와 같은 결과를 얻게 되었는데, 총 55개의 숫자를 오인식하여 2.75%의 오류율과 1853개의 숫자를 정인식하여 92.65%의 인식률을 나타냈으며, 3단계 노드

의 결과 숫자 0과 6에서 잦은 오류를 하였는데, 3개의 노드중에 가장 높은 신뢰도와 인식률을 나타내고 있다.

오류	55 / 2000	2.750(%)
기각	92 / 2000	4.600(%)
정인식	1853 / 2000	92.65(%)
신뢰도	1853 / 1908	97.12(%)

표.3 3단계 파이프라인 인식 결과

본 논문에서 제안하는 병렬 파이프라인의 숫자 별 실험 결과는 표.4, 5와 같이 인식 되었는데, 44개의 숫자를 오인식하여 2.2%의 오류율을 나타냈으며, 63개의 숫자를 기각하여 3.15%의 기각률을 나타내었다. 1893개의 숫자를 정인식하여 94.65%의 인식률을 나타냈으며, 병렬 파이프라인의 신뢰도는 97.73%의 성능을 보였다. 3개의 파이프라인 결과를 비교 했을 때 3단계 노드에서의 오류율이 1~2%정도 낮게 나타났으며, 인식률 면에서는 1~2% 높아져 가장 나은 인식 결과를 보였다.

오류	44 / 2000	2.200(%)
기각	63 / 2000	3.150(%)
정인식	1893 / 2000	94.65(%)
신뢰도	1893 / 1937	97.73(%)

표.4 병렬 파이프라인 인식 결과

3개의 인식기 노드에서의 결과값이 상대적인 오류 숫자의 분포율을 보여 최종 결과에 영향을 주었으며, 결합 단계를 통한 인식 성능의 결과 값이 2~4%정도 높아지는 것을 볼 수 있었으며, 오류율이나 기각률에서도 나은 결과를 보였다.

정인 오류	0	1	2	3	4	5	6	7	8	9	오류	기각	정인식(%)
0	192								1	2	3	5	192 (96.0)
1		194	2					1			3	3	194 (97.0)
2		2	187	1		1	1			1	6	7	187 (93.5)
3				184		2	1		2		5	11	184 (92.0)
4		1			191					1	2	7	191 (95.5)
5			1	2		186	3	1			7	7	186 (93.0)
6	1							190		2	3	7	190 (95.0)
7		2	1						192	1	4	4	192 (96.0)
8	1			1				3		189	1	6	189 (94.5)
9	2			1	1	1				188	5	7	188 (94.0)
Σ											44	63	1893(94.63)

표.5 병렬 파이프라인 숫자별 결과

8. 결론

본 논문에서는 병렬 파이프라인 다중 인식기를 이용한 필기체 숫자를 인식하기 위하여 C#을 가지고 설계/구현 하였다. 조합전에 각 노드별 숫자를 인식하는 성향에서 각기 다른 특성을 보였기 때문에 조합시 인식률의 향상을 도모 할 수 있었다. 이런 성능 향상과 함께 인간의 관점에서 명백하게 판단 가능한 숫자에 대해 인식시의 많은 오류를 개선하였다는 점과 시스템을 구현하였다는데 큰 의미를 갖는다. 본 논문의 검증기는 구조적 특징을 사용하였기 때문에 입력 영상의 잡영이나 왜곡 등의 구조적 결함을 갖고 있을 경우 올바른 검증을 하기 어려웠다. 검증기의 최대 성능을 발휘하기 위해선 입력 영상의 구조적 특성을 보존하면서 잡영이나 왜곡을 처리해주는 효과적인 전처리가 요구된다.

[참고문헌]

- [1] E.Mandler and J.Schuermann, "Combining the classification results of independent classifiers based on the Dempster/Shافر theory of evidence," in *Uncertainty in Artificial Intelligence*, pp.381-393, Elsevier Science Publishers, 1988.
- [2] Y.S.Hung and C.Y.Suen, "An Optimal Method of Combining Multiple Classifiers for Unconstrained Handwritten Numeral Recognition." In *Proceedings of the 3rd International Workshop on Frontiers in Handwriting Recognition*. pp.11-20, 1993.
- [3] J. Franke and E. Mandler, "A Comparison of Two Approaches for Combining the Votes of Cooperating Classifiers." In *Proceedings of the 11th IAPR International Conference on Pattern Recognition*. Vol. 2. pp.611-614, 1992.
- [4] P. Garder, et al, "Recognition of handwritten digits using template and model matching," *Pattern Recognition*, Vol.24, No.5, pp.421-431, 1991.