

템플레이트 매칭 분류를 이용한 SOFM의 분할 학습과 특징 추출

서석배, 하성욱, 강대성
동아대학교 전자공학과 뉴미디어 통신 연구실
부산광역시 사하구 하단동 동아대학교 전자공학과
g9773109@seunghak.donga.ac.kr

Divided SOFM Training and Feature Extraction Using Template Matching Classifier

Serk Bae Ser, Seung Wook Ha, Dae-Seong Kang
Dept. of Electronic Engineering, Dong-A University
840 Hadan-dong, Saha-ku, Pusan 604-714, Korea
Tel: +82-51-200-7710 Fax: +82-51-200-7712 E-mail: g9773109@seunghak.donga.ac.kr

요 약

In this paper, a new algorithm is proposed that the template matching is used to divide SOFM (Self-Organizing Feature Map) for fast learning and to extract features for considering input data types. In order to verify the superiority of the proposed algorithm, applied to the recognition of handwritten numerals. Templates of handwritten numerals are created by a line of external-contact.

을 여러 개의 작은 SOFM으로 분할하는 방법은 입력의 고유적인 특성을 기준으로 한다. 이 기준은 하나의 SOFM으로 학습시 반드시 다른 노드로 분할되는 특성이어야 하며, 이 특성을 잘 못 설정하였을 경우 전체 알고리즘의 성능은 크게 저하된다. 본 알고리즘의 성능을 평가하기 위해서 Concordia 대학에서 제공하는 4000개의 필기체 숫자를 사용하였다. 2000개는 학습을 위한 숫자이며, 나머지 2000개는 성능 평가를 위한 숫자이다. 먼저 전체 학습 필기체 숫자에 대하여 외접선을 발생하여 전체적인 숫자의 형태를 파악한 다음, 그것을 기준으로 기준 템플레이트(template)를 작성하고 각 템플레이트마다 효과적인 특징추출 방법을 적용한다. 그림 1은 본 알고리즘의 개요를 나타낸다.

1. 서론

기존의 SOFM의 경우, 학습 성능은 SOFM의 학습이 끝난 후 평가할 수 있고, 만약 학습 성능이 만족할 수준에 도달하지 못하면 처음부터 다시 학습을 하여야 한다. 따라서 학습시간이 많이 걸릴 뿐만 아니라, 제대로 분류된 패턴의 정보도 사용하지 못한다. SOFM의 특성을 향상시키기 위해서 유전 알고리즘, 신경 회로망, 퍼지 알고리즘 등 많은 알고리즘이 도입되어 학습의 시간과 성능을 대폭 향상시켰지만 SOFM의 근본적인 문제점은 해결하지 못하였다^[1]. 최적의 학습은 수많은 시행착오에 의해서 이루어지며, 특히 입력의 개수와 SOFM의 노드가 많은 경우엔 학습을 더욱 더 어렵게 한다. 그리고 학습의 성능은 SOFM의 노드수와 비례하므로 학습 시간과 학습 성능의 타협점을 찾기가 매우 힘들다^[2-3]. 본 연구에서는 SOFM이 학습될 때 분류되는 특징을 먼저 분할함으로써 하나의 큰 SOFM이 아닌 분할된 여러 개의 SOFM을 단계적으로 학습한다. 학습 성능의 평가는 분할된 여러 개의 SOFM의 학습 성능의 평균으로 측정 가능하며 학습이 잘 되지 않은 SOFM을 재학습함으로써 전체적인 학습률을 높일 수 있다. 하나의 큰 SOFM

2. 전처리

전처리 필터(preprocessing filter)는 입력 필기체 숫자에 대하여 크기와 굵기 그리고 위치의 정보를 정규화 한다. 입력 필기체 숫자는 64×64 화소의 그레이 영상(gray scale image)으로 임계치를 설정하여 주어진 입력과 동일 크기의 이진 영상(binary image)으로 변환한다. 먼저 이진 영상에서 salt and pepper noise 같이 분산되어 있는 작은 크기의 잡음을 제거하기 위해 dilation과 erosion필터를 적용하였다^[4]. 다음으로 잡음을 제거시킨 이진 영상에서 배경을 제외한 숫자 부분을 검출하여 일정한 크기로 확대한다. 숫자 부분 검출에는 숫자에 외접하는 사각형을 숫자의 영역으로 간주하였다. 검출한 숫자 부분을 일정한 크기로 확대하면 입력 숫자의 크기는 정규화되지만 동시에 숫자가 전체적으로 굵어진다. 굵어진 숫자를 일정한 굵기로 하기 위해서 thinning 필터를 적용하였다. 본 실험에서는 필기 숫자의 세선화에 특성이 우수한 에지검출 세선화 알고리즘을 사용하였다^[5]. 이상의 필터에 의하여 입력 숫자는 잡음이 제거되고 위치는 무시되며 일정한 크기의

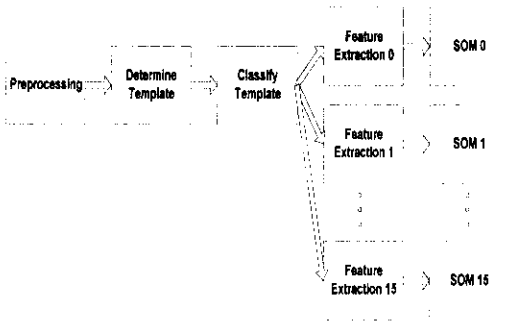


그림 1 전체 블록 다이어그램

크기의 이전 영상이 된다. 그림 2는 전처리 필터의 구성을 나타낸다.

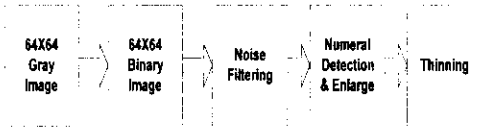


그림 2 전처리 필터의 구성

3. 특징추출과 SOFM의 분할

(a) 입력 필기체 숫자의 형태 파악

모든 형태의 문자에는 일정한 형태가 있다. 비록 쓰는 사람의 글씨체나 글꼴에 따라서 다양한 형태가 존재하지만 전체적인 모양을 보면 유사한 특성을 가지고 있다. 예를 들면 숫자 1의 경우는 면적이 좁은 긴 직사각형 형태나 대각선 형태를, 숫자 0의 경우는 면적이 넓은 사각형이나 원의 형태를 가진다. 본 연구에서는 이 형태를 추출하기 위해서 여러 가지 기울기의 직선을 발생하고, 그 중 숫자 부분과 접하는 접선을 이용하여 전체적인 형태를 결정한다. 그림 3 (a)는 형태결정을 위한 직선 발생을 설명한다. $L_1(n)$ 은 식 (1)과 같이 나타낼 수 있으며 나머지 직선의 방정식 $L_2(n)$, $L_3(n)$, $L_4(n)$ 도 그와 유사하게 표현할 수 있다.

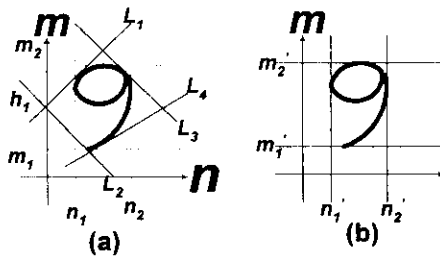


그림 3 숫자 9의 형태 파악

$$L_1(n) = \frac{m_1 - m_2}{n_1} n + h_1 \quad (1)$$

여기서 기울기는 숫자부분인 $f_{num}(n, m)$ 과 접하지 않는 최대의 절대값을 나타낸다. h_1 을 m_1 부터 m_2 까지 감소시키며 접선을 발생하고, 숫자를 포함하며 발생된 접선들로 둘러싸이는 영역을 $R_1(n, m)$ 라고 한다. 다음으로 그림 3 (b)처럼, 상하좌우의 경계를 설정하기 위한 네 개의 직선을 설정하여 각 방향의 경계를 결정하는 n'_1, n'_2, m'_1, m'_2 을 계산한다. 이 접선들은 각 방향에 대하여 숫자영역 $f_{num}(n, m)$ 에 접하지 않는 각 방향의 최대 또는 최소 경계를 나타낸다. 이 네 개의 접선에 의하여 형성되는 영역을 $R_2(n, m)$ 라 한다. 전체적인 숫자의 형태인 $R_s(n, m)$ 은 식 (2)처럼 $R_1(n, m)$ 와 $R_2(n, m)$ 의 교집합으로 표현된다. 그림 4에서 그의 예를 보인다.

$$R_s(n, m) = R_1(n, m) \cap R_2(n, m) \quad (2)$$



$$R_s(n, m) = R_1(n, m) \cap R_2(n, m)$$

그림 4 숫자 9의 형태 결정의 예

(b) 템플레이트 매칭에 의한 패턴 분류

본 연구에서는 Concordia 대학의 2000개의 학습데이터에 대하여 형태 파악 알고리즘을 적용하고 전체적인 모양을 설정한 다음, 그림 5에 나타난 16개의 적절한 템플레이트를 설정하였다. 숫자 입력은 16개의 패턴 중 하나로 분류되어서 특징이 추출되고 각각의 SOFM에 학습된다. 유사한 패턴으로 분류하는 방법으로 유클리드 거리 계산법(Euclidean distance method)을 이용하였다^[6]. 유클리드 거리의 계산식은 식 (3)과 같다. 여기서 $P_i(n, m)$ 은 입력 영상에서 n, m 위치의 화소값이며, $P_j(n, m)$ 은 템플레이트 영상에서 n, m 위치의 화소값으로 0 또는 1의 값을 가진다.

$$EUD = \sum_{n=1}^{64} \sum_{m=1}^{64} (P_i(n, m) - P_j(n, m)) \quad (3)$$

(c) 특징 추출과 SOFM의 학습

가. 특징 추출

입력 필기 숫자는 16개의 템플레이트로 분류된다. 각 템플레이트 영상을 살펴보면 숫자의 정보를 포함할 가능성이 높은 부분과 낮은 부분으로 대별될 수 있다. 그림 5에서 템플레이트 0번의 경우는 중앙의 좁은 영역에 숫자 정보가 존재할 가능성이 높으며 나머지 부분은 가능성이 매우 낮다. 따라서 특징 추출을 위한 메시(mesh)를 설정할 때 중요한 부분은 소밀하게 그리고 정보가 존재할 가능성이 희박한 부분은 간격을 넓게함으로써 특징점의 수를 줄이면서 그 정보를 효과적으로 표현할 수 있다. 그림 6에서는 각 템플레이트의 특징 추출을 위한 메시를 나타내는데, 특징점의 개수가 24에서 56까지 존재한다. 템플레이트 16개를 기준으로 할 때, 기존의 64개

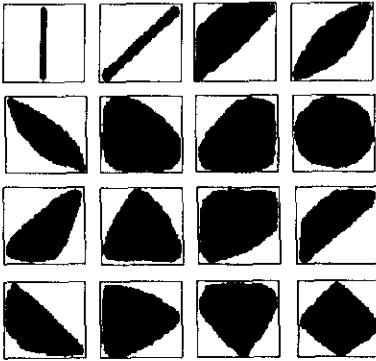


그림 5 실험에 사용한 16개의 템플레이트

로 일정한 특징점을 추출하는 방식에 비하여 26.87% 정도의 메모리가 절약된다.

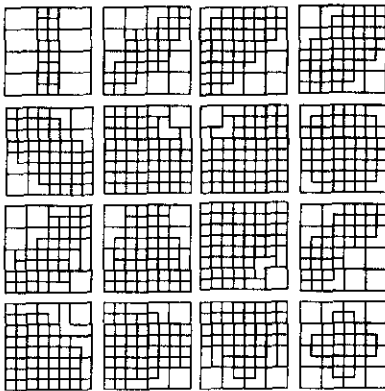


그림 6 특징 추출을 위한 메시들

나. SOFM의 학습

16개의 각 템플레이트에 대하여 각각 SOFM을 적용한다. SOFM은 2차원 형태로, 학습물을 적용할 때 퍼지 알고리즘을 이용하여서 학습의 효율을 높인다[5]. 각 SOFM의 노드의 개수는 식 (4)로 설정한다.

$$Node = (int)\sqrt{N_{template}} \tag{4}$$

여기서, $Node$ 는 각 SOFM의 노드 개수이며 $N_{template}$ 는 각 템플레이트에 분류된 입력 필기숫자의 개수이다. 그리고 (int) 는 계산 결과에 대한 최대의 정수 값을 나타낸다. FSOM의 알고리즘은 다음과 같다.

FSOMstep 1 초기화 :

- a. 연결 강도 $v_{ij}(1 \leq i \leq 64, 1 \leq j \leq m)$ 를 0과 1사이의 값으로 랜덤하게 구성한다. (m : 노드개수)
- b. 최소 오류 $Er > 0.0$ 설정

FSOMstep 2 유클리드 거리 계산 : ($x_i(t)$: 입력패턴)

$$EUD_j = \sum_{i=1}^{64} (x_i(t) - v_{ij}(t))^2$$

FSOMstep 3 승리 노드(winner node), j^* 의 선정

- a. 다음과 같은 최소 유클리드 거리를 갖는 연결 강도와 위치를 구한다.

$$\|X - V_{j^*}\| = \min_j \|X - V_j\|$$

- b. 반복 회수마다 최소 유클리드 거리의 합(sum of winner Euclidean distance: $SWEUD$)을 구한다.

$$SWEUD_j = \sum_{i=1}^{64} EUD_{ij}$$

FSOMstep 4 퍼지 학습률(fuzzy learning rate : u_{ij})의 계산 :

$$u_{ij} = \frac{EUD_{j^*}}{EUD_j} \times \tau(t), \quad u_{ij} = [0.0, 1.0],$$

$\tau(t)$: 감소 계수

FSOMstep 5 연결 강도의 갱신 :

승리 노드를 중심으로 퍼지 소속 정도인 퍼지 학습률과 비례적으로 연결 강도를 다음의 식으로 갱신한다. 갱신되는 범위를 감소한다.

$$v_{ij}(t+1) = v_{ij}(t) + u_{ij}(t)(x_i(t) - v_{ij}(t))$$

FSOMstep 6

IF $SWEUD < Er$ 이면 학습을 종료한다.

Else FSOMstep 2에서 FSOMstep 5까지 반복한다.

4. 실험결과

(a) 전처리

그림 7은 0에서 9까지의 숫자에 대하여 잡음제거, 일정한 크기로 확대, 세선화한 결과이다. 작은 크기의 잡음이 제거되었고 모든 숫자의 크기와 굵기가 일정해졌음을 볼 수 있다.

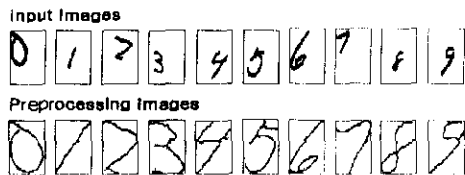


그림 7 세선화 결과

(b) 형태 결정과 템플레이트 분류

그림 8은 학습 숫자의 일부에 대하여 확대, 세선화, 형태 결정, 분류에 의한 특징 추출 결과를 보여준다. 0의 경우는 템플레이트 3으로, 1의 경우는 템플레이트 1로, 2의 경우는 템플레이트 5로, 3의 경우는 템플레이트 4로, 그리고 6의 경우는 템플레이트 15로 분류되었다. 따라서 각각의 메시지를 사용하여 특징값을 추출한다. 그림 8에서 (a)는 원 입력영상, (b)는 확대된 결과, (c)는 세선화 결과, (d)는 형태 결정 결과, (e)는 각각의 메시에 의한 특징추출을 나타낸다. 특징 추출에서 각각 숫자의 정보가

있는 부분은 세밀하게 추출됨을 볼 수 있다.

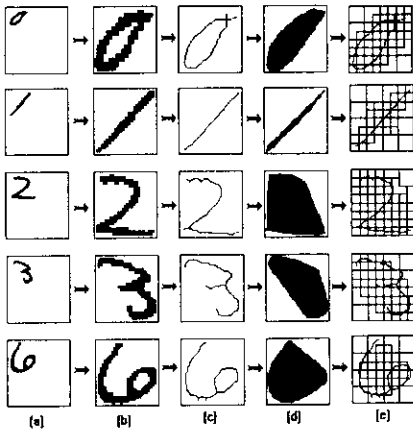


그림 8 형태 결정과 특징 추출

(c) SOFM의 인식률

전체 필기 숫자 데이터의 분류 결과와 각각의 SOFM에 대하여 인식률을 표 1에 나타내었다. 학습은 학습 패턴을 100% 인식할 때까지 반복하며 학습 패턴을 모두 인식하면 테스트 패턴을 시험한다. 여기서 최대 인식률이란 각각의 SOFM에서 학습 패턴을 완벽하게(100%) 학습시킨 후 테스트 패턴을 시험할 때 도달할 수 있는 인식률의 최대 값이다. 최대 인식률은 학습패턴과 테스트 패턴을 템플레이트에 의하여 분류하는 과정 중에 나타나는 오차에 기인한다.

실제 실험 결과, SOFM 0과 SOFM 6은 100%의 인식률을 보였다. 그리고 템플레이트 9의 경우는 최대 인식률이 98.94%이지만 97.87%의 인식률을 보였다.

5. 결론

본 연구에서는 템플레이트 매칭을 이용하여 SOFM을 분할 학습하고 각 SOFM에 대한 특징추출을 달리하는 알고리즘을 제안하였다. SOFM을 분할하여 학습하므로 학습의 효율을 높이며 학습시간을 단축할 수 있었다. 여기서 학습시간의 의미는 1회 학습시간이 아니라 원하는 학습률에 도달할 때까지의 반복에 의한 학습 시간의 총합을 말한다. 알고리즘의 성능을 개선시키기 위해서는 적절한 템플레이트 설정과 학습을 위한 데이터의 선정이 중요한 변수이다. 실험결과를 살펴보면 각 SOFM에 대하여 학습 시 고려되지 않은 숫자가 전체 인식률을 나쁘게 하는 요인으로 작용하였다. 특히 SOFM 13의 경우는 최대 인식률이 75% 밖에 안되는 결과를 초래하였다. 더 좋은 성능을 위해서는 기준 템플레이트를 임의로 설정하지 않고 전체 데이터의 형태에 대한 통계적 특성을 이용하는 방법이 바람직할 것이다.

6. 참고문헌

[1] Zheru Chi, Jing Wu and Hong Yan, "Handwritten Numeral Recognition Using Self-Organizing Maps and Fuzzy Rules," Pattern Recognition, Vol. 28, No. 1, pp. 59-66, 1995.
 [2] T. Kohonen, "The Self-Organizing Map," Proceeding of IEEE 78(9), 1464-1480, 1990.
 [3] Laurence Fausett, "Fundamentals of Neural Networks," Prentice-Hall Inc., 1994.
 [4] Ioannis Pitas, "Digital Image Processing Algorithms," Prentice-Hall Inc., 1993.
 [5] 서석배, 강대성, "신경회로망과 퍼지이론을 이용한 필기 숫자 인식," 생산기술연구소 논문지, 동아대학교부설 생산기술연구소, 1998.2
 [6] J. T. Tou, R. C. Gonzalez, "Pattern Recognition Principles," Addison-Wesley Publishing Company. 1974.

표 1 각 SOFM에 대한 정보

SOFM		각 숫자의 개수										최대 학습률	최대 인식률	
		0	1	2	3	4	5	6	7	8	9			
0	학습		16									1		100%
	테스트	45												
1	학습		99			2						1	2	95.53%
	테스트	77				8	1	1			1	2		
2	학습						13	8			79	1	27	100%
	테스트						14	7			65	3	24	
3	학습	18	76	3	18	30	39	18	7	96	9			100%
	테스트	20	73	3	19	42	34	8	5	85	14			
4	학습		4	2	2	3						2	1	100%
	테스트		2		1	3			1	1	1	2		
5	학습	6	1	43	27				15			2		98.80%
	테스트	2		35	21			6	21			4		
6	학습	55		54	59	1	23	33				9		100%
	테스트	43		35	48	2	27	31				3	1	
7	학습	53		11	33	6	21	10				19	1	99.38%
	테스트	66		12	33	7	16	11				15		
8	학습	29		10	32	5	36	83				11	1	100%
	테스트	29		32	39	15	27	74				12		
9	학습	1		46	4		3	18			1	3		98.94%
	테스트	3		37	3	1	4	26						
10	학습	35		7	19	24	20	2	25	45	34			100%
	테스트	36		6	27	10	19	1	11	38	25			
11	학습		3		2	33	45			41	26	78		100%
	테스트		3			32	36			81	27	79		
12	학습		1	5					1			1		100%
	테스트			7										
13	학습				15		2		2	3				75%
	테스트	1		10	4	2					3			
14	학습			2	3	26	3	1	20	8	46			100%
	테스트			1	5	20	1		29	15	33			
15	학습	3		2	1	13	2	13				1		100%
	테스트			2		24	2	5						