

論文98-35C-7-11

DWW알고리즘을 적용한 고속 가중 FDNN의 설계

(Design of High Speed Weighted FDNN applied DWW Algorithm)

李喆熙*, 卞吾星*, 文成龍*

(Cheul-Hee Lee, Oh-Sung Byun, and Sung-Ryong Moon)

요 약

본 논문에서는 영상처리를 위하여 계층적 구조의 DBNN(Decision Based Neural Network)에 양자화 삼각 퍼지함수를 적용한 FDNN(Fuzzy Decision Neural Network)를 구현한 후, 구현된 FDNN의 하드웨어를 설계하였다. 또한 기준영상과 입력영상을 같은 크기로 정규화하여 두 영상을 이루는 픽셀의 무게를 거리로 환산했다. 이 거리에 의해 영상간의 유사성을 찾아 가장 근사한 값을 선택하는 DWW (Dynamic Weight Warping)알고리즘을 FDNN에 적용하였다. 그래서 영상의 잡음을 제거하고, 동시에 정보의 손실을 최소화하고, 최적의 정보를 얻을 수 있었다. 고속 가중 FDNN의 하드웨어를 COMPASS 툴을 이용하여 설계하였다. 전체회로는 61,000개의 게이트로 구현되었으며, 모의실험을 통하여 FDNN의 우수함을 보였다.

Abstract

In this paper, after we got to realized FDNN(Fuzzy Decision Neural Network) applied the quantization triangularity fuzzy function to DBNN(Decision Based Neural Network) of a hierarchical structure for image process, we could design hardware of the realized FDNN. Also it is normalized the standard image and the input image as the same size. We are applied DWW algorithm which selected the closest value with finding similarity of an interval image by this distance to FDNN. So we could calculated in terms of distance to weight of pixel which composed two image and eliminated the noise of image, minimized the lost of information, obtained the optimal information. It is designed hardware of high speed weighted FDNN using COMPASS tool. Also, the total circuit is realized as gates of 61,000 and could show to superiority of FDNN using the simulation.

I. 서론

지도학습 신경망은 검색단계(retrieving phase)와 학습단계(learning phase)로 나누어 지는데 검색단계는 출력값이 어디에 속하는가를 결정하며, 출력값은

입력값과 판별함수(discriminant function), 즉 가중치(weight)에 의해 결정되며, 학습단계에서의 가중치는 최적의 값을 얻기 위하여 학습되어야 한다^[1-3]. 지도학습에서 지도자는 결정이 올바르게 되어졌는지, 그리고 적절하게 조정이 이루어졌는지를 판별하게 되므로 지도학습은 영상들을 정확하게 분류하기 위한 가중치들을 학습시키고 최상의 판별함수에 의해서 결정 경계치를 결정한다. 이와 같이 지도학습을 이용하여 패턴 분류에 적용되고 있는 DBNN 모델은 선형 퍼셉트론 이론을 기초로 결정 경계치에 의해 선형적으로 이진 분류하여 이 결정 경계치를 결정하는 것이다^[7].

* 正會員, 圓光大學校 電子工學科

(Dept. of Electronic Engineering Wonkwang University)

※ 본 연구는 97년도 교육부 반도체 분야 학술 연구 조성비(ISRC97-E-2032)에 의하여 연구되었음
接受日字:1998年4月20日, 수정완료일:1998年7月2日

11]

본 논문에서는 계층적 구조의 DBNN에 퍼지함수를 적용한 FDNN(Fuzzy Decision Neural Network)¹⁸를 구현한 후, 구현된 FDNN의 하드웨어를 설계한다. DWW(Dynamic Weight Warping)알고리즘을 FDNN에 적용하여 거리에 의해 영상간의 유사성을 찾아 가장 근사한 영상을 선택하도록 함으로써 영상의 잡음을 제거함과 동시에 정보의 손실을 최소화하고 최적의 정보를 얻을 수 있는 고속 가중 FDNN의 하드웨어를 설계하고, 모의실험을 통하여 동작을 검증하고자 한다.

II. DWW알고리즘 원리

DWW 알고리즘은 임의의 입력영상과 일정한 영상 이 저장된 기준영상을 정규화한 후 이들 영상을 분할하여 두 영상사이에 있는 오차를 일련의 조건으로 제한하는 비선형 warping 함수에 의해 영상의 가중치를 비교하여 일치하도록 하는 알고리즘이다^[11]. 이 warping 함수는 가중치 허용오차의 값 D_{wm} 을 최소로 하는 경로에 따라 유동성 있게 결정할 수 있다. 먼저 비교대상인 기준영상 R과 입력영상 U의 특징벡터를 범위가 각각 J, I인 벡터로 표시하면 식 (1), (2)과 같다.

$$R = \{ r_1, r_2, r_3, \dots, r_j, r_j \} \tag{1}$$

$$U = \{ u_1, u_2, u_3, \dots, u_i, u_i \} \tag{2}$$

이들에서 동일한 특징 파라미터를 추출한다면 이들 사이의 면적오차는 $i-j$ 평면상의 일련의 점들로써 식 (3)과 같이 표현할 수 있다.

$$w_p = c(1), c(2), \dots, c(k) \tag{3}$$

여기서, $c(k) = (i(k), j(k))$

$i-j$ 평면상의 점들의 집합은 입력영상 U를 기준영상 R에 1:1 매핑(mapping)하는 함수를 표현하므로 이를 warping 함수라 한다. 만일 두 영상사이에 오차가 없다면 warping 함수는 대각선 $i=j$ 와 일치하며, 오차가 커지면 이 함수의 점들은 대각선으로부터 벗어나게 된다. 두 특징벡터 성분 u_i 와 r_j 의 값의 차이는 이들간의 거리에 의해 얻어질 수 있으며 거리는 식(4)에 의해 구할 수 있다.

$$d(c(k)) = d(i(k), j(k)) = d(i, j) = |u_i - r_j| \tag{4}$$

식 (4)과 warping 함수 w_p 를 이용하여 식 (5)와 같은 거리의 가중 합을 표시할 수 있으며, 이는 warping 함수의 양호성을 나타내는 양으로 사용될 수 있다.

$$S(w_p) = \sum_{k=1}^K d(c(k)) \cdot w(k) = \sum_{k=1}^K |u_i - r_j| \cdot w(k) \tag{5}$$

여기서, $w(k)$ 는 음이 아닌 양의 가중계수

식 (5)의 가중거리 합 $S(w_p)$ 는 warping 함수 w_p 가 오차를 최적상태로 조절할 경우 극부적으로 최소값을 갖는다. 이러한 최소값은 두 영상사이의 오차를 조절 한 후에 여전히 남아 있는 기준영상 R과 입력영상 U 사이의 거리라고 볼 수 있고, 오차축 변동에 대하여 대체적으로 일정한 크기를 갖으며, 그 절대치는 제한 범위를 벗어나지 않는 허용 오차 안에 존재하여야 한다^[5, 10]. 이러한 점을 고려한 두 영상 R과 U 사이의 정규화 거리 $D_{an}(R, U)$ 는 식 (6)과 같다.

$$D_{an}(R, U) = \text{Min}_{w_p} \frac{\sum_{k=1}^K d(c(k)) \cdot w(k)}{\sum_{k=1}^K w(k)} \tag{6}$$

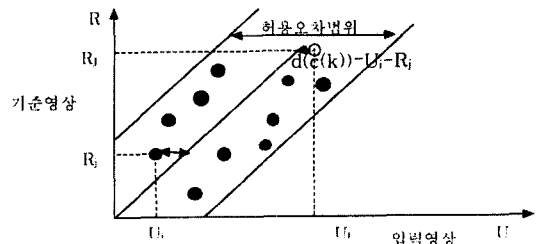


그림 1. DWW 알고리즘 블록도
Fig. 1. DWW Algorithm Block.

식 (6)에서의 오차는 warping 함수와 가중인자를 어떻게 정하느냐에 따라 크게 달라질 수 있다. 그러므로, 바람직한 결과를 얻기 위해서는 취급할 영상의 오차축을 어떻게 표현하느냐에 따라 변화할 것이다. 가중계수는 $c(k)$ 의 좌표값 사이의 거리를 정규화시키는 성분으로 warping 함수에 의해 i, j 축상에 나타나는 $c(k)$ 의 좌표값의 총수에 의한 영향을 보정하기 위한 것으로 정규화거리 $D_{an}(R, U)$ 가 최대값이 되려면 이 warping 함수의 총합을 정규화 시켜 1로 만들면 된다. 식 (6)의 거리함수를 warping 함수 w_p 에 관해

서 최소화하는 문제는 매우 어렵다. 만일 식 (6)의 분모항, 즉 가중계수의 총합을 식 (7)과 같이 N이라 하면

$$N = \sum_{k=1}^K w(k) \quad (7)$$

N 값이 warping 함수 w_p 에 무관한 상수 값이라면 식 (6)은 식 (8)으로 표현할 수 있다.

$$D_{an}(R, U) = \frac{1}{N} \min_w \left[\sum_{k=1}^K d(C(k)) \cdot w(k) \right] \quad (8)$$

예를 들어 7 X 5숫자 패턴에 대하여 패턴 인식에 적용할 경우 DWW에서는 그림 2와 같이 숫자에 대한 한 열을 비교하여 그 면적을 계산후 전체에 대하여 면적을 계산하고 식 (6)으로 거리를 계산하고, 이를 결과에 반영하게 된다. 결과로 알 수 있는 것은 입력패턴과 가장 유사한 기준 패턴을 승리자로 출력하게 된다.

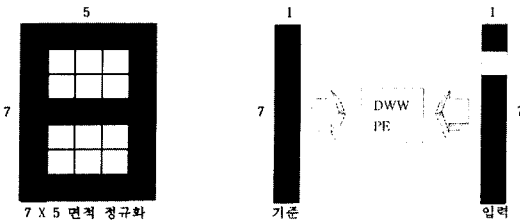


그림 2. 패턴의 면적 정규화 및 연산
Fig. 2. Area Normalization and Operation of Pattern.

III. FDNN의 구성

1. DBNN의 알고리즘

지도학습(supervised learning) 네트워크는 검색단계(retrieving phase)와 학습단계(learning phase)로 나누어지는데 학습단계에서는 영상들을 정확하게 분류하기 위한 가중치들을 훈련시키고 판별함수(discriminant function)에 의해서 결정 경계치(decision boundary)를 결정한다^[7, 14].

신경망 L클래스($\Omega_i, i=1, \dots, L$)중에서 일치되는 $s=(X^{(1)}, X^{(2)}, \dots, X^{(M)})$ 은 주어진 입력영상의 집합이라고 가정하면, 각 클래스는 식별함수 $\phi(X, W_i)$ 을 갖는 하부 망(subnet)으로 모델화 된다. 따라서, 입력영상에 대한 Winning 클래스는 j 번째

클래스(하부망)가 되며 영상에 대하여, $j=i$ 이면 영상 $X^{(m)}$ 은 이미 정확하게 분류되었기 때문에 수정할 필요가 없고, $j \neq i$ 이면 $X^{(m)}$ 은 아직 분류되지 않았기 때문에 식 (9)의 학습조건에 의해서 수정되어야 한다.

$$\begin{aligned} W_i^{(m+1)} &= W_i^{(m)} + \eta \nabla \phi(X, W_i) : \text{긍정학습} \\ W_j^{(m+1)} &= W_j^{(m)} - \eta \nabla \phi(X, W_j) : \text{부정학습} \end{aligned} \quad (9)$$

식 (9)의 DBNN 학습규칙에 퍼지이론을 적용하기 위하여 변형하면 식 (10)과 같으며,

$$\begin{aligned} W_{s_i, q}^{(m+1)} &= W_{s_i, q}^{(m)} + \eta_m \nu_i \pi_i \\ W_{s_j, q}^{(m+1)} &= W_{s_j, q}^{(m)} - \eta_m \nu_j \pi_j \end{aligned} \quad (10)$$

여기서 $\pi_i = \nabla \phi(x, W_i)$ 이고 ν_i 는 L-R 형태 멤버쉽 함수의 파라미터이다.

2. FDNN의 알고리즘

1) 삼각 퍼지함수의 양자화

식 (11)에서 ν_i 는 L-R함수의 멤버쉽 함수로 사용된 파라미터, $[m, \alpha, \beta]_{LR}$ 이고, 노이즈 필터링을 위한 퍼지 평가부를 생산하기 위해 사용된 멤버쉽 함수도 이와 같다. 멤버쉽 함수는 아래와 같다.

$$\nu_{LR-i}(x) = \begin{cases} L\left(\frac{m_l - x}{\alpha}\right) & \text{for } x \leq m_l \\ 1 & \text{for } m_l \leq x \leq m_r \\ R\left(\frac{x - m_r}{\beta}\right) & \text{for } x \geq m_r \end{cases} \quad (11)$$

여기에서 L과 R은 형태(shape)함수이고, 그리고 α, β, ν 는 퍼지 구간에 대한 파라미터이다. L과 R이 가우시안 모양을 갖는다고 가정한다면 식(12)와 같다^[8, 9].

$$\nu_{LR-i}(x) = LR \left[\frac{m_l \nabla x}{\alpha} + \frac{x \nabla m_r}{\beta} \right] \quad (12)$$

여기에서 경계가 다른 연산은 $a \nabla b = \max(a-b, 0)$ 이고, 퍼지 구간은 $I = [m_l, m_r, \alpha, \beta]_{LR}$ 로 나타낸다.

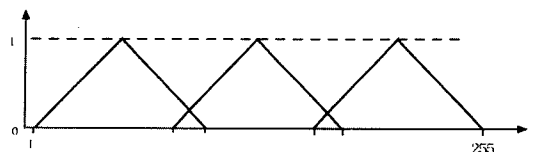


그림 3. 삼각퍼지함수
Fig. 3. Triangularity Fuzzy Function.

본 논문에서는 일반적인 L-R 퍼지 삼각함수를 퍼지화 하기 위하여 그림 3과 같이 퍼지함수를 사용하였다. 그림 3은 L-R 퍼지 삼각함수의 이론을 FDNN에 적용하기 위한 삼각 퍼지함수로서 0 ~ 255사이의 임의의 픽셀값을 [0, 1]로 퍼지화 하는데는 사용된 퍼지함수로 DARK, MIDDLE, BRIGHT의 퍼지함수를 이용하여 퍼지화 하였다. 구간은 DARK(45, 44, 44), MIDDLE(125, 44, 44), BRIGHT(205, 44, 44)으로 설정을 하였으며, WFM(Weighted Fuzzy Means)의 원리를 기초로 그림 3에서 3부분(DARK, MIDDLE, BRIGHT)의 멤버십 함수의 각 영역을 디지털화하기 위해서 1 ~ 89까지 5단위씩 세분화하여서, 그레이 스케일의 픽셀값이 0 ~ 8사이에서 존재하도록 퍼지함수를 양자화 하여 병렬 연산처리가 가능하도록 디지털화 하였다.

2) 가중 FDNN의 구조

그림 3은 삼각함수에 퍼지 개념을 도입한 것으로 전체 0~255를 0과 1사이의 값으로 나타낸 값을 도식화 한 것이다. 만약 I가 그림 3에 저장된 퍼지 구간이라면, $y_E(\cdot)$ 에 의해서 나타나는 퍼지 평가자 E는 다음 식 (13)에 의해서 산출된다.

$$y_E(X(i, j)) = \frac{X(i, j) * x(i, j)}{N_{LR-I}}$$

$$= \frac{\sum_{k=-\frac{n_1-1}{2}}^{\frac{n_1-1}{2}} \sum_{l=-\frac{n_2-1}{2}}^{\frac{n_2-1}{2}} \nu_{LR-I}(x(i+k, j+l)) * x(i+k, j+l)}{\sum_{k=-\frac{n_1-1}{2}}^{\frac{n_1-1}{2}} \sum_{l=-\frac{n_2-1}{2}}^{\frac{n_2-1}{2}} \nu_{LR-I}(x(i+k, j+l))}$$

(13)

여기서 WFM의 $h(i, j)$ 는 $n_1 \times n_2$ 샘플행렬에 의한 임펄스 응답형식인 컨벌루션(convolution) $h(i, j) * x(i, j)$ 를 나타내며, 식 (14)에서 Sugeno-type 퍼지규칙의 퍼지평균을 나타낸다^[13-15].

IF

$x(i-n_1, j-n_2) \text{은 } f, x(i-n_1, j-n_2+D) \text{은 } f, \dots, x(i-n_1, j+n_2) \text{은 } f,$
 $x(i-n_1+I, j-n_2) \text{은 } f, x(i-n_1+I, j-n_2+D) \text{은 } f, \dots,$
 $x(i-n_1+I, j+n_2) \text{은 } f$
 \vdots
 $x(i+n_1, j-n_2) \text{은 } f, x(i+n_1, j-n_2+D) \text{은 } f, \dots, x(i+n_1, j+n_2) \text{은 } f$

THEN

$$\bar{y}(i, j) = \frac{h(i, j) * x(i, j)}{N_i}$$

$$= \frac{\sum_{k=-n_1}^n \sum_{l=-n_2}^m \mu(x(i+k, j+l)) * x(i+k, j+l)}{\sum_{k=-n_1}^n \sum_{l=-n_2}^m \mu(x(i+k, j+l))}$$

(14)

μ 는 입력 픽셀의 퍼지값이고 x 는 입력값이다. 그림 4는 FDNN의 연산과정을 나타낸 것으로 퍼지화된 픽셀 값은 비퍼지화 과정을 통하여 퍼지화된 데이터에서 원래 값을 찾기 위해서 5x5 행렬의 25개의 픽셀 값을 퍼지화하고 퍼지평균은 식(14)와 같으며 퍼지 평균에서 L-R 퍼지집합인 비퍼지화에 대한 규칙의 추론 결과를 계산한다. 퍼지평균에 의해서 얻어진 값은 식(13)을 이용하여 오차가 가장 적은 출력 값을 얻기 위한 평가자 값을 구하게 된다^[16].

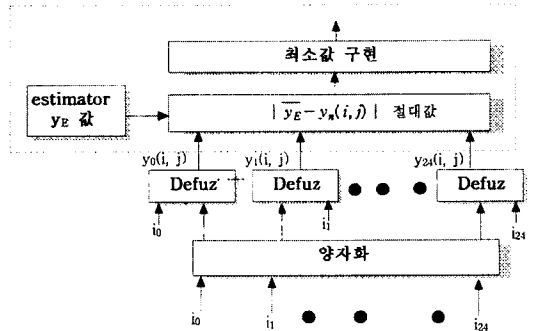


그림 4. FDNN의 연산과정
Fig. 4. Operation process of FDNN.

IV. 고속가중 FDNN의 하드웨어 구현

1. FUZZIFY 블록

그림 5는 일반적인 L-R 퍼지 셀로써 구성은 식 (12)로부터 유추되며, WFM 필터에서 사용한 t-norm과 s-norm연산을 소거함으로써, MIN과 MAX 연산과정을 제거하였다.

이 각 모듈은 $n_1 \times n_2$ 잡음 화소의 퍼지집합을 산출하기 위해 윈도우를 적용하였다. 그림 5는 퍼지화 블록으로써 8비트 입력 값은 퍼지함수(DARK, MIDDLE, BRIGHT)를 이용하여 4 비트로 양자화 된다. 양자화는 삼각 퍼지함수에서 값과 일치하도록 하였으며, 양자화를 거친 데이터는 평가자 블록과 비퍼지화 블록으로 전송된다.

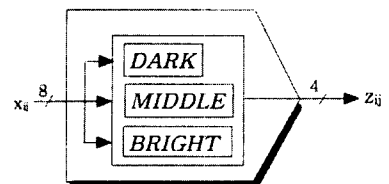


그림 5. 퍼지화 블록도
Fig. 5. Fuzzify Block.

2. 비퍼지화 블록

퍼지 평균은 $\frac{\sum \sum \text{입력} \times \text{퍼지값}}{\sum \sum \text{퍼지값}}$ 이므로 X_n 과 μ_n 을 곱하여 더한 후 μ_n 을 더한 값으로 나눈다. 이 과정에서 3x3행렬에 대한 비퍼지값을 얻게되고 이 값들은 decision블록에 보내진다. 비퍼지화 과정에서는 5x5행렬을 이용한 25개의 픽셀 값을 퍼지 평균 연산을 이용하여 평가자를 구하게 된다. 그리고 퍼지화 과정과 비퍼지화과정은 병렬로 구현하여 실시간 처리에 접근 하도록 설계되었으며, 그림 6과 같다.

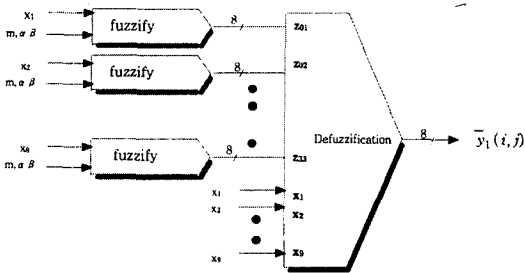


그림 6. 비퍼지화 블록도
Fig. 6. Defuzzification Block.

그림 6의 비퍼지화 블록은 2개의 블록으로 설계하였으며, 평가자와 서브 행렬의 계산 방법은 같다. 평가자에서는 25개의 픽셀값과 양자화 값을 연산하고 서브 행렬에서는 9개의 픽셀을 처리하게 된다.

모의실험에서 평가자블록은 242ns, 비퍼지화 블록은 226ns의 지연시간이 발생하였으며, 효율적인 클럭사용과 동작시간 단축을 위해서 다운-에지 트리거(down-edge trigger) 모드로 동작하도록 설계하였다.

3. DECISION 블록

비퍼지화된 데이터들은 $\bar{y}_E - \bar{y}_n(i, j)$ 의 절대값을 계산하고 식별 값이 첨가된 후 MIN블록에서 최소 값을 찾는다. 식별 값으로 S블록에서는 $\bar{y}_1(i, j) \sim \bar{y}_9(i, j)$ 값 중 \bar{y}_E 와 최소 값을 갖는 비퍼지 값을 출력하고 이 데이터는 DWW에서 \bar{y}_E 의 오차가 지정한 범위의 값을 만족할 때까지 반복 연산하여 출력된다.

그림 7은 decision블록의 블록도 이고, 입력을 받은 9개의 비퍼지화 값은 1~9까지의 식별 값이 첨가된다. 식별값은 스위칭 블록에서 최소값을 갖는 $\bar{y}_n(i, j)$ 가 선택되어야 함으로 스위칭 블록의 디코더에서 스위칭 입력으로 사용된다. 이렇게 입력된 8비트 값은 식별 값 4비트가 첨가되어 $|\bar{y}_E - \bar{y}_n(i, j)|$ 가 계산되고,

나머지와 캐리가 발생하는데, 나머지는 MIN 블록에서 최소값 연산이 되고 그 데이터의 식별 값은 스위칭 블록의 디코더에 입력되어 최소값을 갖는 $\bar{y}_n(i, j)$ 을 출력한다. 캐리는 DWW 블록에서 최소값의 수정에 사용된다.

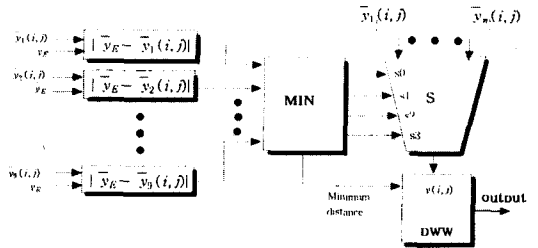


그림 7. DECISION 블록도
Fig. 7. DECISION Block.

MIN블록에서 MIN회로는 8개를 사용하였으며, 몇 번째값이 최소인가를 판단하게되고 실제값의 식별값만을 디코더 블록으로 넘겨주게 된다. MIN블록에서의 최종 결과는 최소값을 갖는 식별값이다.

4. DWW 블록

DWW블록은 기준영상과 입력영상을 같은 크기로 하여 두 영상을 이루는 픽셀의 무게를 거리로 환산하여 이 거리에 의한 영상간의 오차를 수정할 수 있도록 설계하였다. 그림 8은 DWW블록의 블록도를 나타낸 것이다. 먼저 \bar{y}_E 와 $\bar{y}_n(i, j)$ 을 비교하여 오차를 검사하고, 오차이내이면 현재값을 오차가 범위를 벗어난 경우에는 수정된 값을 출력한다. 수정과정은 8비트 A에서 dist [4:0]을 뺄셈연산 후 캐리를 검사하고 절대값 연산에서 발생한 캐리정보를 가지고 ADD/SUB를 결정하여 연산한 후 수정된 값을 출력한다. 값의 출력은 MUX에서 결정하게되며 스위칭은 오차를 검사하는 블록(-dist)의 캐리에 의해서 동작한다.

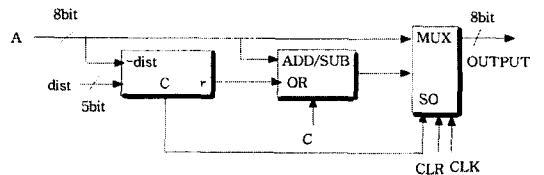


그림 8. DWW 블록도
Fig. 8. DWW Block.

5. 전체 FDNN 하드웨어

그림 9는 DWW 알고리즘을 적용한 FDNN의 회로

로써, 영상정보 8비트의 데이터는 레지스터 블록에서 5 X 5원도우형태로 입력되고, 클럭 동기에 의하여 다운 에지(down-edge)에서 25개의 데이터가 평가자(estimator), 비퍼지화 블록과 퍼지화 블록으로 전송된다. 퍼지화 블록에서는 8비트 데이터를 4비트로 퍼지화 하고 퍼지화된 4비트 데이터는 평가자(estimator)블록, 비퍼지화 블록에 입력된 후 퍼지 평균 연산을 한다. 평균 연산은 병렬로 처리되고, 연산된 결과는 다음 클럭의 다운 에지(down-edge)에서 decision 블록으로 전송된다.

비교하여 선정된 오차 이내로 수정되어 최종 픽셀 정보를 출력하게 된다. 그림 9의 전체 하드웨어는 퍼지화(FUZZIFY)블록, 비퍼지화(DEFUZZIFICATION)블록, DECISION 블록, 평가자(estimator)블록 4부분으로 구성하였다.

6. 모의 실험 및 결과

그림 12는 원 영상에 대한 히스토그램을 나타낸 것이고, 그림 13~15는 입력영상에 30%, 50%, 70%의 잡음이 포함된 영상에 대하여 WFM과 FDNN을 적용한 결과의 히스토그램을 나타낸 것으로 FDNN이

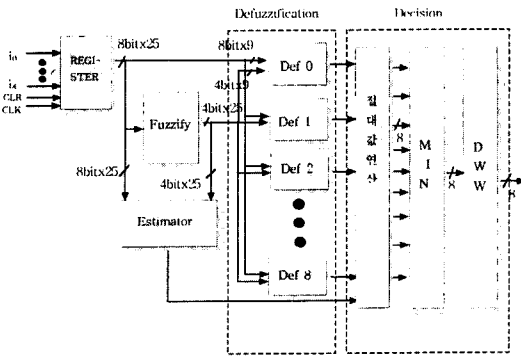


그림 9. 전체 FDNN의 하드웨어 블록도
Fig. 9. Hardware Block of the entire FDNN.

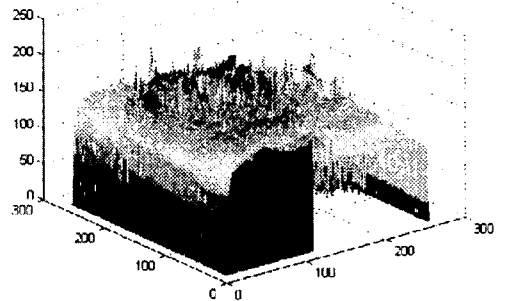


그림 12. 원영상의 히스토그램
Fig. 12. Histogram of the Original Image.

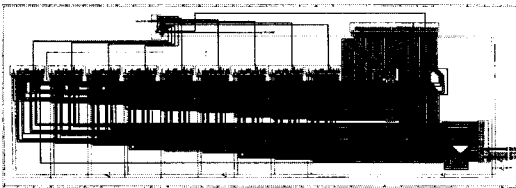
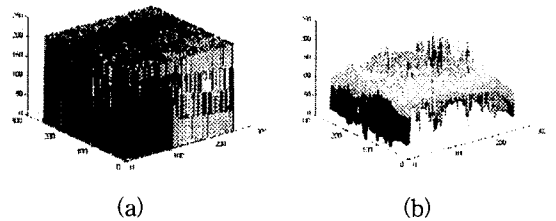


그림 10. 전체 FDNN의 회로도
Fig. 10. Circuit of the entire FDNN.



(a) (b)

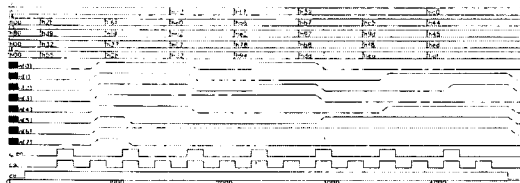
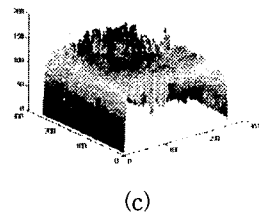


그림 11. 전체 FDNN의 결과 파형
Fig. 11. Result wave of the entire FDNN.



(c)

그림 13. 각 알고리즘에 30% noise 영상의 히스토그램
(a) 30% noise 영상의 히스토그램 (b) 30% noise의 WFM 영상의 히스토그램 (c) 30% noise의 FDNN 영상의 히스토그램
Fig. 13. Histogram of 30% noise in each Algorithm. (a) Histogram of 30% noise Image (b) Histogram of 30% noise with WFM Image (c) Histogram of 30% noise with FDNN Image

Decision 블록에서는 비퍼지화한 9개의 픽셀 값과 평가자와 비교하여 최소 오차 값을 갖는 비퍼지값을 MIN 블록에서 찾는다. DWW 블록에서 평가자 값과

우수한 결과를 얻었으며, 그림 16은 삼각 퍼지함수와 양자화된 삼각 퍼지함수에 의한 WFM과 FDNN을 적용한 결과에 대한 MSE를 나타낸 것으로 FDNN이 원 영상에 가장 근사한 값을 얻을 수 있었다.

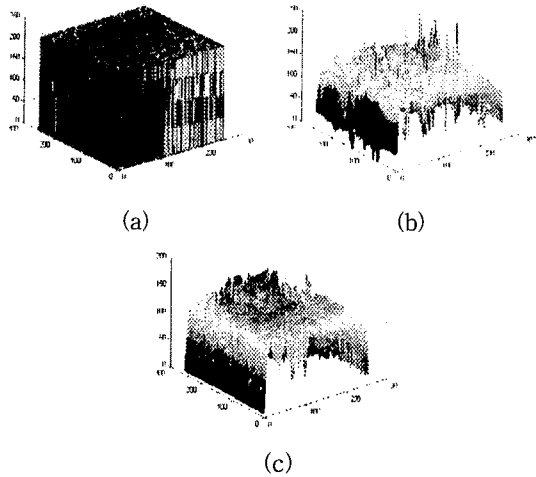


그림 14. 각 알고리즘에 50% noise 영상의 히스토그램 (a) 50% noise의 영상의 히스토그램 (b) 50% noise의 WFM 영상의 히스토그램 (c) 50% noise의 FDNN 영상의 히스토그램
 Fig. 14. Histogram of 50% noise Image in each Algorithm. (a) Histogram of 50% noise Image (b) Histogram of 50% noise with WFM Image (c) Histogram of 50% noise with FDNN Image

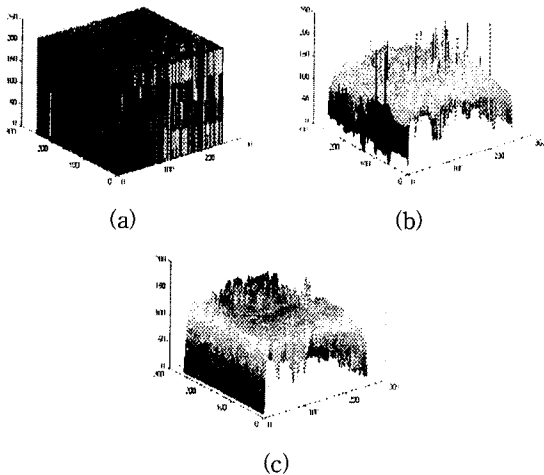


그림 15. 각 알고리즘에 70% noise 영상의 히스토그램 (a) 70% noise 영상의 히스토그램 (b) 70% noise의 WFM 영상의 히스토그램 (c) 70% noise의 FDNN 영상의 히스토그램
 Fig. 15. Histogram of 70% noise Image in each Algorithm. (a) Histogram of 70% noise Image (b) Histogram of 70% noise with WFM Image (c) Histogram of 70% noise with FDNN Image

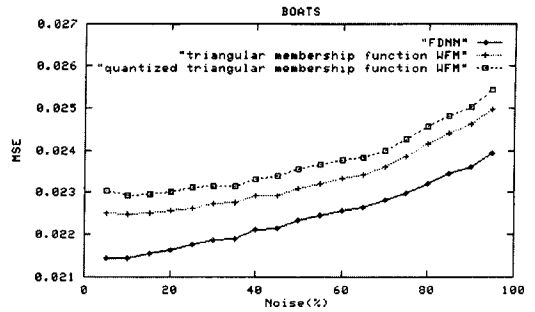


그림 16. Boats 영상에 대한 MSE 곡선
 Fig. 16. Curve MSE for Boats Image.

VI. 결 론

모든 정보 전달이 컴퓨터 네트워크에 의해 이루어지는 정보화 사회에서 정보 누출에 대한 보안유지가 중요하게 대두되고 있으며, 개인식별, 보안 시스템 등의 응용 분야에 적용할 수 있는 인식 시스템의 개발이 요구됨에 따라서 실시간 처리를 위해 병렬연산 방식과 회로의 디지털화가 요구되고 있다. 계층적 구조의 DBNN은 임의의 입력 데이터가 입력되면 기준 영상 중에서 유사도가 가장 큰 기준 영상을 찾아 국부승리자로 선택하고, 국부 승리자 중에서 전체 승리자를 선택하여 최종 출력한다.

본 논문에서는 영상처리를 위한 계층적 구조의 DBNN알고리즘에 양자화 삼각 퍼지함수를 적용하여 FDNN(Fuzzy Decision Neural Network)를 구현하였으며, 구현된 FDNN(Fuzzy Decision Neural Network)의 하드웨어를 설계하였다.

또한 기준영상과 입력영상을 같은 크기로 정규화하여 두 영상을 이루는 픽셀의 무게를 거리로 환산하고, 이 거리에 의해 영상간의 유사성을 찾아 가장 근사한 값을 선택하는 DWW(Dynamic Weight Warping) 알고리즘을 FDNN에 적용하여 영상의 잡음을 제거함과 동시에 정보의 손실을 최소화하고 최적의 정보를 얻을 수 있는 고속 가중 FDNN의 하드웨어를 COMPASS 틀을 이용하여 설계하였으며, 전체FDNN 회로는 61,000개의 게이트로 구현되었다.

원 영상에 30%, 50%, 70%의 잡음이 포함된 영상에 대하여 WFM, 고속 가중 FDNN을 적용하여 모의 실험한 결과 그림 12~15에서 나타난 바와 같이 FDNN이 원영상에 가장 가까운 영상을 얻을 수 있음을 보였다.

앞으로 처리속도를 보다 향상시킴으로써 동영상, 각종 보안 시스템 등에 응용할 수 있을 것으로 사료된다.

참 고 문 헌

[1] Robert L. Harvey, Neural Network Principles, Prentice-all, Inc., 1994.

[2] S. Y. Kung, Digital Neural Networks, PTR Prentice-Hall, Inc, 1993.

[3] K. Fukushima, "A Neural Networks for Visual Patten Recognition", IEEE Computer Magagine, pp. 65-75, March, 1988.

[4] K. Wojtek Przytula, Viktor K. Prasanna, Parallel Digital Implementations of Neural Networks, Prentice-Hall, Inc, 1993.

[5] R. C. Gonzalez, R. E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, 1992.

[6] S. Y. Kung, J. N. Hwang, "Parallel architecture for artificial neural nets", In Processding IEEE; International Conference on Neural Networks, vol. 2, pp. 165-172, July 1988.

[7] S. Y. Kung, J. S. Taur. Decision based neural networks with signal/image classification application. In Proceeding, IEEE Transations on neural, networks

for signal Processing, 1993

[8] C. H. Chen, Fuzzy Logic and Neural Network Handbook, McGraw-Hill Series on computer Engineering, 1997.

[9] 이광형, 오길록, 퍼지이론 및 응용(I, II권), 홍릉과학출판사, 1991

[10] M. A. Sid-Ahmed, Image Processing, McGraw-Hill, Inc. 1995.

[11] 송창영, 문성룡, 김환용, 패턴인식을 위한 디지털 DBNN의 설계, vol. 21/ no. 11, 한국통신학회논문지, Nov, 1996

[12] BART KOSKO, Fuzzy Engineering, Prentice Hall International, Inc. 1997.

[13] Chao-Lieh Chen, Chang-Shing Lee and Yau-Hwang Kuo, Design of high speed weighted Fuzzy mean filters with generic LR fuzzy cells, IEEE, 1996.

[14] Bernji, Hamid R. Fuzzy logic and neural networks, Piscataway, NJ; Institute of Electrical and Electronics Engineers, Inc, 1992.

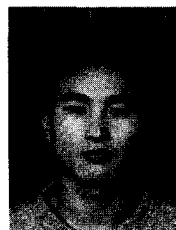
[15] Granino A. Korn, Neural networks and fuzzy-logic control on personal computers and workstation, Cambridge, Mass; MIT Press, 1995.

[16] 최선곤, 문성룡, "양자화 삼각 퍼지함수를 적용한 FDNN 구현 및 비교 분석", 대한 전자공학회 논문 (접수)

저 자 소 개



李 喆 熙(正會員)
 1998년 2월 22일 원광대학교 전자공학과 졸업(공학사). 현재 원광대학교 대학원 전자공학과 석사과정. 관심분야는 비선형 회로 및 신경망 시스템 설계, 패턴인식, 영상신호처리, VLSI 설계



卞 冪 星(正會員)
 1997년 2월 22일 원광대학교 전자공학과 졸업(공학사). 현재 원광대학교 대학원 전자공학과 석사과정. 관심분야는 비선형 회로 및 신경망 시스템 설계, 패턴인식, 영상신호처리, VLSI 설계

文 成 龍(正會員) 第 35卷 S編 第 4號 參照
 원광대학교 전자공학과 조교수