

다층구조 퍼셉트론을 이용한 분류 영상압축 및 코딩

正會員 趙 廣 輔* 正會員 朴 哲 勳** 正會員 李 庸 水**

Classified Image Compression and Coding using Multi-Layer Perceptron

Kwang Bo Cho*, Soo Young Lee**, *Regular Members*

요 약

본 논문에서는 블록 분류와 코딩과 함께 신경회로망을 이용한 영상압축을 보인다. 오차 역전파 알고리즘으로 학습되는 다층구조 신경회로망은 정규화된 영상데이터를 감소된 공간 중복성을 가지는 은닉층의 값으로 변환하는데 사용된다. 기본적으로 영상압축은 입력층과 출력층의 뉴런보다 적은 수의 은닉층 뉴런에 의해 얻어진다. 여기에 시각체계의 민감도에 따른 영상블록의 복잡성에 따라 적응적으로 압축되도록 블록을 분류한다. 또한 은닉뉴런의 양자화된 값은 효과적인 전송을 위해 entropy coding이 된다. 컴퓨터 모의실험에서 학습된 영상뿐만 아니라 테스트 영상에 대해서도 좋은 성능을 얻었으며 entropy coding을 이용한 경우 화질의 큰 저하없이 약 25:1의 압축률을 얻었다.

ABSTRACT

In this paper, image compression based on neural networks is presented with block classification and coding. Multilayer neural networks with error back-propagation learning algorithm are used to transform the normalized image data into the compressed hidden values by reducing spatial redundancies. Image compression can basically be achieved with smaller number of hidden neurons than the numbers of input and output neurons. Additionally, the image blocks can be grouped for adaptive compression rates depending on the characteristics of the complexity of the blocks in accordance with the sensitivity of the human visual system(HVS.) The quantized output of the hidden neuron can also be entropy coded for an efficient transmission. In computer simulation, this approach lie in the good performances even with images outside the training set and about 25:1 compression rate was achieved using the entropy coding without much degradation of the reconstructed images.

* 금성중앙연구소

Goldstar Central Research Laboratory

** 한국과학기술원 전기 및 전자공학과

Dept. of EE., Korea Advanced Institute of Science and Technology(KAIST).

論文番號 : 94206

接受日字 : 1994年 7月 30日

I. 서론

영상압축(Image Compression)은 영상을 표현할 때 필요한 비트들의 수를 줄이고, 이를 이용하여 원래 영상으로 복원하는 것이다. 영상과 같이 데이터 양이 많고 적용 분야가 많을수록 효과적인 영상압축은 매우 중요하다. 현재 사용되고 있는 정지 영상압축의 국제표준으로 되어 있는 JPEG(Joint Photographic Expert Group)⁽¹⁾은 기본적으로 Discrete Cosine Transform(DCT)을 이용하여 영상압축하는 것으로, 압축률이 약 20~30:1까지 좋은 성능을 나타내고 있으나 그 이상의 높은 압축률은 알고리즘에 한계가 있다.

또 다른 대표적인 것으로 Vector Quantization(VQ)⁽²⁾이 있다. 이는 영상블럭을 하나의 벡터로 보고 이를 코드화하여 영상압축하고자 하는 것으로 높은 압축률을 얻을 수 있다. 그러나 성능에서는 JPEG이 좋은 성능을 보이는 영역에서는 JPEG보다 떨어진다.

한편 최근 많은 연구가 진행중인 신경회로망은 많은 계산량을 요구하는 신호 처리분야에 효과적으로 적용할 수 있는데, 이는 신경회로망이 갖고 있는 본질적인 대규모 병렬성(massive parallelism), 적응학습(adaptive learning), 일반화(generalization) 특성에 기인한다⁽³⁾. 따라서 많은 양의 데이터를 다루어야 하는 영상분야에서는 중요한 장점으로 작용한다. 그 예로 다층구조 신경회로망은 영상데이터를 압축된 데이터인 은닉층의 출력으로 변환시킴으로 영상이 갖고 있는 공간적 중복성(spatial redundancy)을 제거한다^(4,5,6,7). 여기서 신경회로망이 영상압축에 쓰일 수 있는 장점을 Sonehara등⁽⁴⁾은 다음과 같다고 하였다.

- (1) 원래의 데이터를 축소된 특징 공간에 매핑하는 데이터 압축 특성.
- (2) 학습된 데이터뿐만 아니라 학습되지 않은 데이터에 대해서도 정보 처리를 제공하는 일반화 특성.

신경회로망을 영상압축에 적용한 예를 들면 다음과 같다. 다층 신경회로망, 특히 2층 신경회로망을 이용한 예로, Sonehara등⁽⁴⁾은 하이퍼큐브 병렬 컴퓨터로 DCT를 이용하여 신경회로망의 초기 연결강도(weight)를 DCT 계수로하여 영상압축하였고, Marsi등⁽⁵⁾은 DCT를 이용하여 블럭을 나누고, 이들 각각의 코드를 만들어 영상압축을 하였으며, Arduini등⁽⁶⁾은 입력 영상블럭을

class로 나누고 블럭크기를 16×16, 8×8로 나누어 adaptive하게 처리하였다. Namphol등⁽⁸⁾은 영상압축에 입력으로 입력의 higher order term을 사용했다.

다른 신경회로망을 적용한 경우를 보면, Liu등⁽⁹⁾은 Counter Propagation Neural Network을 이용하여 영상압축을 하였으며, Russo등⁽¹⁰⁾은 Outer Product Neural Networks(OPNN)이라는 모델을 제안하여 이것을 이용하여 영상압축을 하였고, Gedeon등⁽¹¹⁾은 Pruning Trained Network를 써서 은닉층의 수를 최소화하였다. 또한 Chen등⁽¹²⁾은 self organizing feature map을 이용하여 적응적 VQ하는데 이용하였고 Lancini등⁽¹³⁾은 VQ 코드북(codebook)을 설계하는데 신경회로망을 이용하였다. 그러나 전체적으로 아직까지 성능에 있어 기존의 방법보다 떨어지며, 이에 대한 체계적인 비교가 없어 정확한 비교를 하기 힘들다.

본 논문에서는 신경회로망을 이용해서 영상압축을 하는 것으로, 신경회로망만을 이용하여 영상압축하는 것과 성능을 향상시키기 위하여 HVS를 고려한 분류 알고리즘을 사용하여 영상 블럭을 4개의 클래스로 나누어 적응적 압축을 하는 방법을 제안한다. 또한 압축 데이터의 효과적인 전송과 압축률을 높이기 위해 entropy coding을 부가적으로 사용한다. 이를 기존 방법인 JPEG, VQ와 비교하여 신경회로망을 이용한 영상압축의 성능을 보인다.

논문의 전체 구성은 2장에서는 영상압축에 이용하는 신경회로망을 보이며, 3장에서는 제안한 영상압축 및 복원과정을 보이며, 4장에서는 시뮬레이션 결과 및 비교분석을, 마지막으로 5장에서 결론을 맺도록 한다.

II. 영상압축에 이용하는 신경회로망

다층구조 신경회로망(MLP : Multi-layer Perceptron)

다층구조 신경회로망은 원래 영상의 중복성(redundancy)을 줄이기 위해 사용된다. 신경회로망의 학습은 일반적으로 많이 쓰이는 오류 역전파(error back propagation)알고리즘⁽¹⁴⁾을 사용하여 출력값이 입력값과 같아지도록 학습한다. 적용할때 은닉층에 있는 뉴런수는 그림1과 같이 입력과 출력층의 뉴런수보다 적어야 한다. 데이터압축과 구현의 용이성을 위해서 단순한 2층 신경회로망을 사용하였다⁽⁷⁾.

한 픽셀당 256 gray 레벨을 가지는 512×512 영상을 입력뉴런의 수보다 적은 수의 은닉뉴런을 가지는 2층 신경회로망에 입력하기 위해서 8×8픽셀 블럭으로 나

는다. 따라서 입력과 출력층의 뉴런수는 64개이나 그림 1에서 보듯이 적응적 임계값(adaptive threshold)을 이용하기 위해 1인 값을 가지는 입력뉴런이 하나 더 있고 학습시 은닉층의 뉴런은 양자화되어 학습된다. 은닉층의 출력값이 압축된 데이터로, 입력영상 데이터에 대해 입력뉴런수와 은닉뉴런수의 비, 입력영상의 양자화값과 은닉뉴런출력의 양자화값의 비로 압축된다. 압축하는 것은 영상블럭을 입력층에서 은닉층으로 전달함으로써 은닉층의 뉴런값들로 압축되고, 영상블럭을 복원하는 것은 은닉층의 뉴런값들을 은닉층에서 출력층으로 전달함으로써 실행된다.

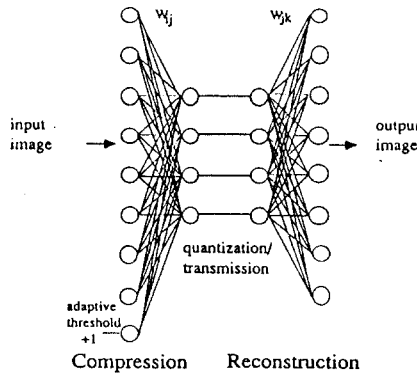


그림 1. 영상 압축에 이용되는 신경회로망 구조
Fig. 1. Neural network structure for image compression.

III. 영상압축 및 복원 과정

제안한 영상압축 및 복원 과정은 그림2와 같다. 전체적으로 살펴보면, 압축과정은 먼저 영상을 8×8 픽셀 크기의 블럭으로 나누어서 전처리 과정으로 블럭을 분류한다. 다음 신경회로망의 입력으로 정규화하고 은닉층의 값으로 양자화되어 압축된다. 압축률을 더 높이기 위해 entropy coding을 사용한다.

복원과정은 코딩된 신호를 디코딩(decoding)하여 그 값을 신경회로망의 후단을 통과하여 최종적으로 복원된 영상을 보인다. 블럭 경계면이 보이는 blocking effect가 생길 때는 필터를 이용하여 이를 제거할 수 있다.

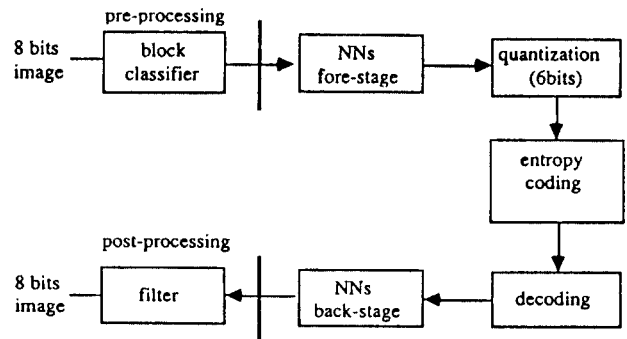


그림 2. 영상 압축 및 복원 과정
Fig. 2. Image compression/reconstruction process

1. 전처리

영상의 복잡도(complexity)를 고려하여 블럭을 분류하고, 복잡도에 따라 2층 구조 신경회로망의 은닉층의 수를 다르게 하여 인간의 시각체계의 민감도에 따라 적응적으로 영상압축을 할 수 있게 하는 단계로, 영상 분류 알고리즘은 Ramanurthi등¹⁵⁾이 사용한 알고리즘을 수정해서 사용한다. 간단히 설명하면, 수평, 수직 방향의 gradient로 인접한 픽셀간의 gray 레벨 차이를 이용하여 블럭내의 edge 성분이 얼마나 있는가에 따라 블럭을 클래스로 나누는 것으로, 먼저 영상을 N×N 블럭으로 나누고 수평, 수직 방향의 gradient로 인접한 픽셀간의 gray 레벨 차이를 정규화한 값을 shade와 edge 성분으로 구분하는 임계값(threshold) T_e , T_s 을 정하고 그 블럭내 각 픽셀 간에 이 값이 얼마나 있는지를 나타내는 카운터가 있어 카운터 값을 이용하여 실험에 의해 정해진 임계값 J_e , J_s 에 의해 네 블럭으로 구분된다.

분류 알고리즘(classification algorithm)

각 블럭에 대하여 다음과 같은 순서로 수행되어 블럭을 나눈다.

- i) 크기가 N×N인 블럭 X의 (i, j)번째를 X_{ij} 라 한다
- ii) 다음과 같은 6개의 카운터와 2개의 임계값(threshold)을 둔다.
6개의 카운터 : S_{11} , S_{12} , H_{11} , V_{11} , V_{12} = 0로 한다.
2개의 임계값 : T_e (edge threshold), T_s (shade threshold)¹⁵⁾은 다음 식과 같다.

$$T_e = \begin{cases} 8.0/d_{av} & \text{if } d_{av}(\text{average gray level}) < 30.0 \\ 0.2 & \text{otherwise} \end{cases} \quad (1)$$

$$T_s = \begin{cases} 0.1 & \text{if } d_{av} < 30.0 \text{ or } d_{av} > 225.0 \\ 0.25 & \text{otherwise} \end{cases} \quad (2)$$

iii) 다음 식 (3)은 수평, 수직 방향의 gradient로 인접한 픽셀간의 gray레벨 차이를 정규화한 것이다.

$$d_h = \frac{2(x_{i,j} - x_{i,j+1})}{x_{i,j} + x_{i,j+1}}, d_v = \frac{2(x_{i,j} - x_{i+1,j})}{x_{i,j} + x_{i+1,j}} \quad (3)$$

iv) 카운터는 다음 식들과 같이 된다. 여기서 h(horizontal), v(vertical), p(positive), n(negative)이다.

$$\begin{aligned} S_h < S_{h+1} & \text{ if } |d_h| > T_s, \text{ for } i=1,2,\dots,N, j=1,2,\dots,N-1 \\ S_v < S_{v+1} & \text{ if } |d_v| > T_s, \text{ for } i=1,2,\dots,N-1, j=1,2,\dots,N \\ H < H_{h+1} & \text{ if } d_h > T_c, \text{ for } i=1,2,\dots,N, j=1,2,\dots,N-1 \\ H_n < H_{n+1} & \text{ if } d_h < -T_c, \text{ for } i=1,2,\dots,N-1, j=1,2,\dots,N \\ V_p < V_{p+1} & \text{ if } d_v > T_c, \text{ for } i=1,2,\dots,N, j=1,2,\dots,N-1 \\ V_n < V_{n+1} & \text{ if } d_v < -T_c, \text{ for } i=1,2,\dots,N-1, j=1,2,\dots,N \end{aligned} \quad (4)$$

v) 블럭 분류(classification)

다음 식과 같은 임계값 J_s (shade block), J_e (edge block)를 설정하여 블럭을 분류한다.

다음 식과 같은 임계값 J_s (shade block), J_e (edge block)를 설정하여 블럭을 분류한다.

$$J_s = N-1, J_e = N/2 \quad (5)$$

if $S_h < J_s$, and $S_v < J_s$, then Shade block (class 1)

else if ($V_p \geq J_e$ and $V_n \geq J_e$) or ($H_p \geq J_e$ and $H_n \geq J_e$)

Complicated block (class 2)

else if ($V_p \geq J_e$ or $V_n \geq J_e$) or ($H_p \geq J_e$ or $H_n \geq J_e$)

Edged block (class 3)

else Midrange block (class 4)

이 알고리즘을 사용하여 영상블럭은 4개의 클래스로 구분된다. 즉 Shade, Complicated, Edged, Midrange 클래스로 구분되는데, Lena 영상에서 블럭을 분류한 예를 그림 3에 보았다. 여기서 Shade 블럭은 블럭전체가

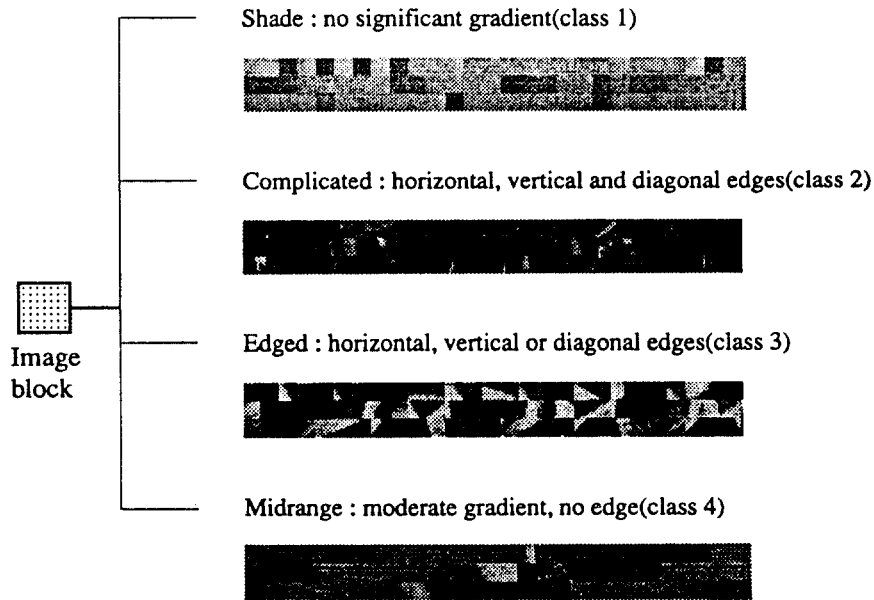


그림 3. 분류된 블럭 예

Fig. 3. Examples of the classified blocks.

눈으로는 gray 레벨이 평균값으로 보이는 것으로 압축 시 평균값만 보내면 된다. 이는 웨버의 법칙(Weber's law)을 이용한 것이다. 웨버 법칙을 간단히 설명하면, 어느 한 픽셀이 주위 픽셀(배경)과 차이가 Weber fraction 이상 차이가 나지 않으면 인간이 그 픽셀들이 같은 영상레벨 값을 갖는 것으로 보는 인지심리학적 실험을 통해 얻어진 법칙이다. 위의 분류 알고리즘에서 T가 Weber fraction을 단순화한 것이다. Complicated 블록은 영상특성이 가장 복잡한 것으로 수평, 수직, 대각선 edge들이 섞여 있는 블록이고, Edged 블록은 수평, 수직, 대각선 edge중 한 형태만 존재하는 것이다. 나머지 Midrange 블록은 edge가 없고 Shade 블록이 아닌 것으로, Shade와 Midrange 블록의 구분은 주위 배경에 비해 밝기 차이가 Weber fraction보다 작고 큰에 따라 분류된다^(15,16).

2. 매핑(mapping)

신경회로망에 학습시키기 위해 입력값의 동적범위를 정규화한다. 실험에서는 픽셀당 0~255레벨을 1~1로 정규화하였다.

그림 4는 매핑 구조를 나타내는 것으로, 정규화된 영상블록은 신경회로망의 입력층의 입력으로 되고, 출력층에서 복원된 영상블록을 얻을 수 있다. 여기서 Shade 클래스는 신경회로망이 필요없고 Complicated 클래스는 8개의 은닉 뉴런, Edged 클래스는 6개의 은닉 뉴런, Midrange 클래스는 4개의 은닉 뉴런을 각각 사용했다.

3. 양자화(quantization)

전송된 계수값의 무제한성을 고려하여 데이터의 압축을 할때 계수들을 한정된 데이터 길이를 갖는 제한된 수의 값들로 할당하는 방법을 양자화라 한다. 양자화에는 uniform 양자화와 non-uniform 양자화가 있다. 이 과정에서 양자화 잡음이 발생하므로 lossy coding이다. 실험에서는 단순한 uniform 양자화를 사용하였는데 이는 변환영역에 대해서 동등한 간격의 값들을 할당한다. 신경회로망의 은닉층 출력이 1에서 1이므로 이를 6비트로 양자화하여 보낸다⁽¹¹⁾.

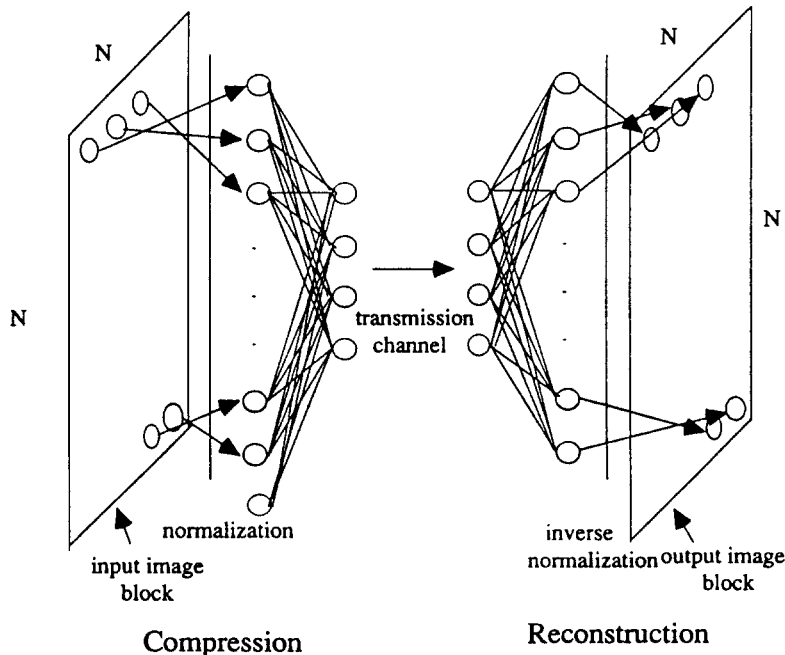


그림 4. 매핑구조

Fig. 4. Structure for the mapping stage.

4. Entropy Coding

양자화된 계수는 최종적으로, 통계적 특성에 따라 가변장부호(variable length code)를 사용한 entropy coding을 통해 더욱 압축된다. 이 과정은 양자화 과정과는 달리 손실이 없는 loseless coding이다. Entropy code로는 Huffman code, arithmetic code, universal code 등으로 주로 Huffman code를 많이 사용하는데, Huffman coding의 lower bound인 entropy H 를 식(6)에 보이겠다⁽¹⁷⁾.

$$H = - \sum_{i=1}^L p_i \log_2 P_i \quad (6)$$

P_i 는 i 번째 계수가 일어날 확률이다. 여기서 $\sum_{i=1}^L p_i = 1$ 이므로 $0 \leq H \leq \log_2 L$ 이다.

정보이론에서 entropy가 이론적으로 전송에 필요한 비트량을 나타내므로, 이를 시뮬레이션에서 이용하겠다. 따라서 직접 코드를 설계한 것은 아니다.

5. 후처리

Nonoverlapping 변환부호화의 최대의 단점은 블럭의 경계가 눈에 보이기 쉬운 소위 blocking effect인데, 이를 줄이기 위해 보통 블럭 경계면에 lowpass 필터를 이용한다. 시뮬레이션에서 블럭 분류기를 이용하는 경우는 blocking effect가 높은 압축률일 때 생기는데, 다음 식(7)과 같은 lowpass 필터를 블럭 경계면에 사용할 수 있다⁽¹⁶⁾.

$$3 \times 3\text{-point } h(n_1, n_2) = \begin{cases} h(0, 0) = 1/5 \\ 1/10 \text{ at the remaining eight points} \end{cases} \quad (7)$$

시뮬레이션에서는 lowpass 필터를 사용하지 않았다. 그러나 사용하면 영상의 질이 좋아질 것이다.

IV. 시뮬레이션 결과 및 비교분석

1. 충실도(성능 비교 기준)와 압축률

영상에서 각 픽셀은 m 개의 비트로 된 2^m 의 gray 레벨값을 가진다. 영상의 객관적인 비교를 위해서 에러측정은 식(8)의 $N \times N$ 입력영상과 $N \times N$ 출력영상 사이의 MSE(mean-square-error)와 식(9)의 peak 신호대 잡음비(peak signal to noise ratio : PSNR)로 나타낸다.

$$MSE = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y) \quad (8)$$

$$PSNR = 10 \log \left(\frac{255^2}{\sum e^2} \right) \quad (9)$$

압축률은 블럭 분류를 하지않고 압축시킬 때는 다음 식 (10), 은닉층의 뉴런 출력값을 entropy coding하는 것을 고려하면 식 (11)과 같이 나타낼 수 있다. 다음 식에서 N^2 은 블럭 크기이며 H 는 은닉뉴런의 갯수이다. N_h 는 입력영상의 픽셀 비트수이고 N_h 는 은닉층 뉴런의 비트수이다.

$$CR = \frac{1}{block \#} \sum_{block} \left(\frac{N^2}{H} \right) \left(\frac{N_i}{N_h} \right) \quad (10)$$

$$CR_{entropy} = \frac{1}{block \#} \sum_{block} \left(\frac{N^2}{H} \right) \left(\frac{N_i}{entropy} \right) \quad (11)$$

블럭 분류기를 사용한 경우 다음 식 (12)과 식 (13)로 나타난다. 블럭 분류기를 사용한 경우는 각 블럭당 $\log_2 class\#$ 의 추가비트가 더 들어간다. 즉 블럭을 네 class로 나누었기 때문에 블럭당 2비트씩 더 소요된다.

$$CR = \frac{1}{block \#} \sum_{block} \left(\frac{N^2 \times N_i}{H_{class} \times N_h + \log_2 class \#} \right) \quad (12)$$

$$CR_{entropy} = \frac{1}{block \#} \sum_{block} \left(\frac{N^2 \times H_i}{H_{class} \times entropy + \log_2 class \#} \right) \quad (13)$$

II. 시뮬레이션 결과

사용된 영상으로는 그림 5에서 보인 256 gray 레벨을 갖고 크기가 512×512 인 Lena, Bridge, Boat, Train 영상을 사용하였다. 전체 시뮬레이션은 학습할 영상을 8×8 블럭으로 나누어 학습하고 그것을 학습된 영상과 다른 영상에 적용하였다.

먼저 전처리를 하지않고 그림 1과 같이 간단한 2층 구조 신경회로망만을 사용하여 Lena 영상을 은닉뉴런 수를 16, 8, 6, 4로 다르게 학습하여 나온 압축률과 화질을 표1에 나타내었다. 여기서 은닉뉴런은 6비트로 양자화되었다.

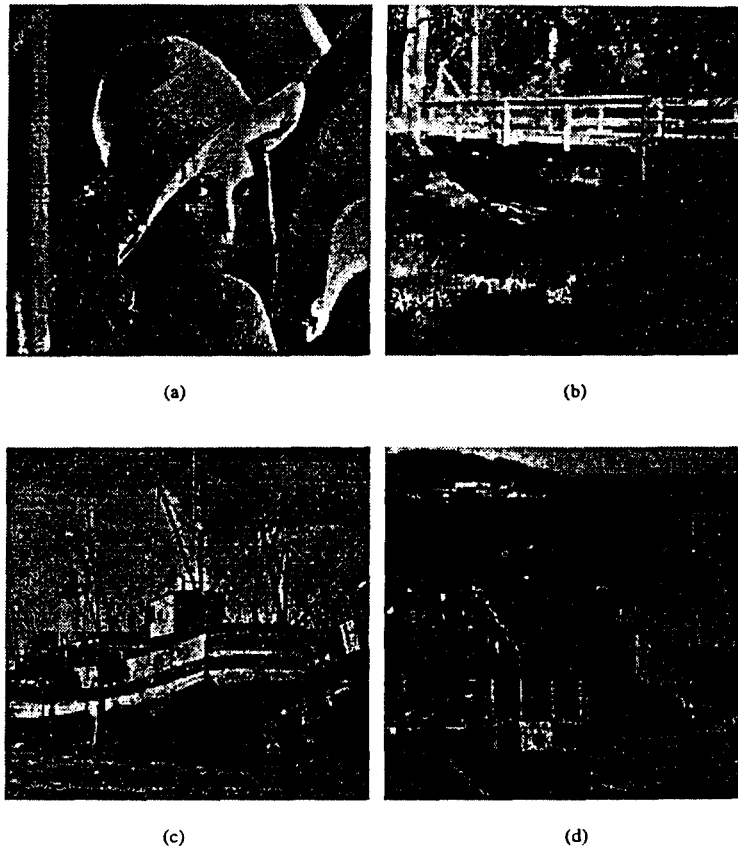


그림 5 시뮬레이션에서 사용한 영상

Fig. 5. Original images in simulation

(a) Lena (b) Bridge (c) Boat (d) Train

이 Lena영상으로 학습된 신경회로망을 가지고 학습하지 않은 영상에 적용한 것으로 표 2는 Bridge 영상에 대해, 표 3은 Boat 영상에 대해, 표 4는 Train 영상에 대해 각각 적용한 결과이고 그림 6은 은닉 뉴런수가 8개, 6개일 때의 Lena와 Boat 영상의 복원된 영상이다.

다음은 블럭 분류기를 사용하여 블럭을 분류한 후에 신경회로망을 각 블럭에 적용적으로 학습시킨 것으로, Shade 클래스(class 1)는 그 블럭의 gray 평균값만 보이면 되므로 신경회로망으로 학습할 필요가 없고, Complicated 클래스(class 2)는 8개의 은닉 뉴런을 갖는 구조로 학습하고, Edged 클래스(class 3)는 6개의 은

Table 1. Performance and compression ratio of trained Lena image without preprocessing

표 1. 전처리를 하지않고 학습한 Lena 영상의 복원 충실도 및 압축률

Neural Network (input-hidden-output)	MSE	PSNR (dB)	entropy	Compression Ratio without entropy coding (CR)	Compression Ratio with entropy coding (CR _{entropy})
65 - 16 - 64	46.559	31.451	3.373	5.33 : 1 (1.50 bpp)	9.48 : 1 (0.84 bpp)
65 - 8 - 64	57.809	30.511	3.543	10.67 : 1 (0.75 bpp)	18.07 : 1 (0.44 bpp)
65 - 6 - 64	66.989	29.871	3.658	14.23 : 1 (0.56 bpp)	23.31 : 1 (0.34 bpp)
65 - 4 - 64	98.012	28.218	4.289	21.33 : 1 (0.38 bpp)	29.84 : 1 (0.27 bpp)

표 2. 전처리를 하지않고 학습하지 않은 Bridge 영상의 복원 충실도 및 압축률

Table 2. Performance and compression ratio of untrained Bride image without preprocessing.

Neural Network (input-hidden-output)	MSE	PSNR (dB)	entropy	Compression Ratio without entropy coding (CR)	Compression Ratio with entropy coding (CR _{entropy})
65 - 16 - 64	211.569	24.876	3.635	5.33 : 1 (1.50 bpp)	8.80 : 1 (0.91 bpp)
65 - 8 - 64	263.516	23.923	3.894	10.67 : 1 (0.75 bpp)	16.44 : 1 (0.49 bpp)
65 - 6 - 64	298.849	23.3761	3.906	14.23 : 1 (0.56 bpp)	21.86 : 1 (0.37 bpp)
65 - 4 - 64	356.822	22.618	4.598	21.33 : 1 (0.38 bpp)	29.84 : 1 (0.27 bpp)

표 3. 전처리를 하지않고 학습한 Lena 영상의 복원 충실도 및 압축률

Table 3. Performance and compression ratio of untrained Bride image without preprocessing.

Neural Network (input-hidden-output)	MSE	PSNR (dB)	entropy	Compression Ratio without entropy coding (CR)	Compression Ratio with entropy coding (CR _{entropy})
65 - 16 - 64	66.449	29.906	3.431	5.33 : 1 (1.50 bpp)	9.32 : 1 (0.86 bpp)
65 - 8 - 64	89.289	28.623	3.550	10.67 : 1 (0.75 bpp)	18.03 : 1 (0.44 bpp)
65 - 6 - 64	106.045	27.876	3.611	14.23 : 1 (0.56 bpp)	23.64 : 1 (0.34 bpp)
65 - 4 - 64	151.395	26.330	4.273	21.33 : 1 (0.38 bpp)	29.95 : 1 (0.27 bpp)

표 4. 전처리를 하지않고 학습하지 않은 Train 영상의 복원 충실도 및 압축률

Table 4. Performance and compression ratio of untrained Bride image without preprocessing.

Neural Network (input-hidden-output)	MSE	PSNR (dB)	entropy	Compression Ratio without entropy coding (CR)	Compression Ratio with entropy coding (CR _{entropy})
65 - 16 - 64	29.064	33.497	3.644	5.33 : 1 (1.50 bpp)	8.78 : 1 (0.91 bpp)
65 - 8 - 64	39.404	32.175	3.664	10.67 : 1 (0.75 bpp)	17.47 : 1 (0.46 bpp)
65 - 6 - 64	47.168	31.394	3.723	14.23 : 1 (0.56 bpp)	22.93 : 1 (0.35 bpp)
65 - 4 - 64	75.474	29.353	4.360	21.33 : 1 (0.38 bpp)	29.35 : 1 (0.27 bpp)

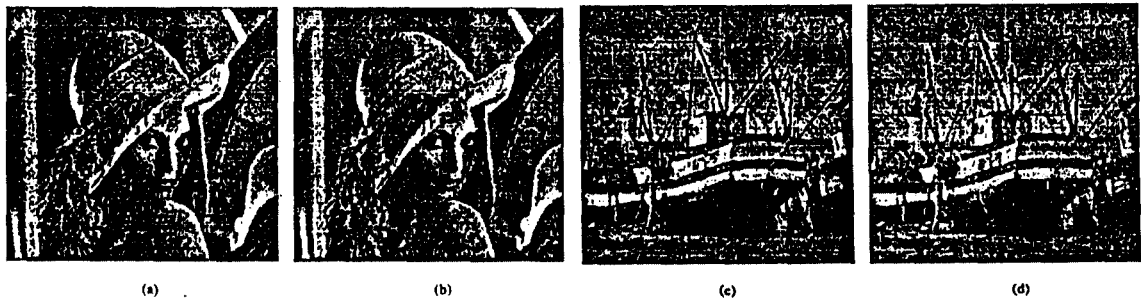


그림 6. 전처리를 하지않고 신경회로망과 entropy coding을 사용하여 복원된 영상

Fig. 6. Reconstructed image using neural network and entropy coding without preprocessing.

- (a) Lena (Hidden neuron : 8, PSNR : 30.511, CR : 18.07 : 1)
- (b) Lena (Hidden neuron : 4, PSNR : 28.218, CR : 29.84 : 1)
- (c) Boat (Hidden neuron : 8, PSNR : 28.623, CR : 18.03 : 1)
- (d) Boat (Hidden neuron : 4, PSNR : 26.330, CR : 29.95 : 1)

표 5. 블럭 분류 알고리즘을 이용한 각 영상의 클래스 분포

Table 5. Class distribution of each image using block classification algorithm.

image \ class	Lena	Bridge	Boat	Train
1 (shade)	259 (6.3%)	46 (1.1%)	541 (13.2%)	203 (4.9%)
2 (Complicated)	586 (14.3%)	2366 (57.8%)	814 (19.9%)	339 (8.3%)
3 (Edged)	576 (14.1%)	734 (17.9%)	829 (20.2%)	987 (24.1%)
4 (Midranged)	2675 (65.3%)	950 (23.2%)	1912 (46.7%)	2567 (62.7%)

닉 뉴런을 갖는 구조로 학습했으며, Midrange 클래스(class 4)는 4개의 은닉 뉴런을 갖는 구조로 학습하였다.

물론 이것이 각 클래스를 대표하는 구조는 아니며 적당한 충실도와 압축률을 얻기 위해 은닉층의 뉴런수를 조정할 수 있다.

각 영상에 대해 블럭분류 알고리즘을 적용하여 4개의 클래스로 나눈 결과를 표 5에 보였다. 표 5에서 보듯이 Bridge 영상은 Complicated 블럭이 전체 블럭에서 약 58%를 차지하고 있는 굉장히 복잡한 영상임을 알 수 있다. 따라서 이 영상을 압축하여 복원할 경우 그 성능이 많이 저하될 것을 알 수 있으며, 이 경우는 영상블럭을 더 나누면 좋은 효과를 기대할 수 있다. 복원된 학습영상으로 Lena 영상을 갖고 학습시켜 복원시킨 결과와 다른 영상에 적용한 결과를 표 6과 그림 7에 보였다.

표 6. 분류된 Lena 영상블럭으로 학습하고 다른 영상에 적용한 결과

Table 6. Results for reconstructed images with learning by the classified blocks of Lena.

Image	Mean Square Error (MSE)	Peak Signal to Noise Ratio (PSNR)dB	Entropy	Compression Ratio without entropy coding (CR)	Compression Ratio with entropy coding (CR _{entropy})
Trained Lena	59.89	30.37	3.972	18.05:1 (0.44bpp)	27.12:1 (0.29bpp)
Bridge	270.20	23.814	3.938	12.77:1 (0.63bpp)	12.77:1 (0.63bpp)
Boat	94.59	28.372	3.988	17.45 (0.46bpp)	26.43:1 (0.30bpp)
Train	48.95	31.233	4.099	18.07:1 (0.44bpp)	26.51:1 (0.30bpp)

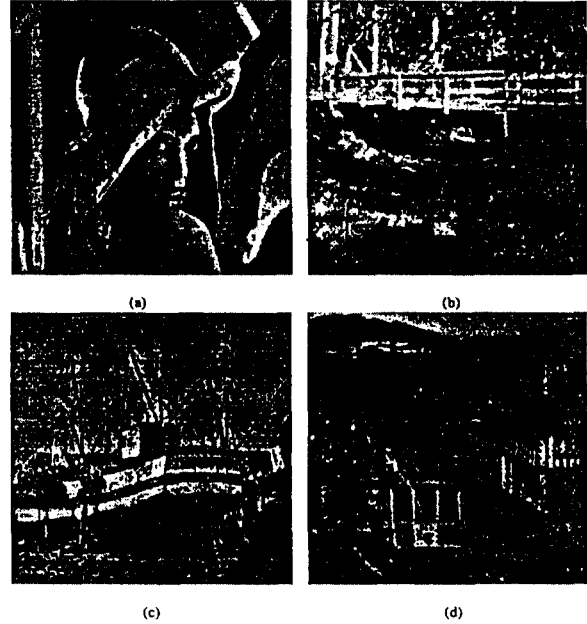


그림7. 블럭 분류기와 entropy coding을 이용하여 복원된 영상

Fig. 7. Reconstructed image using block classifier and entropy coding.

(a) Lena (PSNR : 30.357, CR : 27.12:1)

(b) Bridge (PSNR : 23.814, CR : 19.41:1)

(c) Boat (PSNR : 28.372, CR : 26.43:1)

(d) Train (PSNR : 31.233, CR : 26.51:1)

III. 비교 분석

지금까지의 결과에서 보듯이 학습하지 않은 영상에 대해서도 좋은 결과를 보임을 알 수 있다. 다음 그림 8은 기존의 영상압축 알고리즘과 비교하기 위하여 Lena 영상을 사용한 결과이다.

기존의 영상압축 알고리즘으로는 정지영상의 국제 표준인 JPEG, VQ로는 비교적 좋은 성능을 보이는 Interpolative/Residual VQ(I/RVQ)⁽¹⁷⁾을 사용하였다. 간단히 I/RVQ에 대해 설명하면, 영상블럭을 sub-sampling하여 복원하고 원래 영상에서 복원된 영상을 뺀 나머지 영상을 VQ를 이용해 코딩하는 것으로 blocking effect가 생기지 않고 압축률과 성능이 비교적 좋게 나오는 방법이다.

그림 8에서 N은 neural networks, E는 entropy coding, B는 block classifier을 각각 나타낸다. 한 예로 BNE는 블럭 분류기와 신경회로망, 그리고 entropy coding을 사용한 것이다.

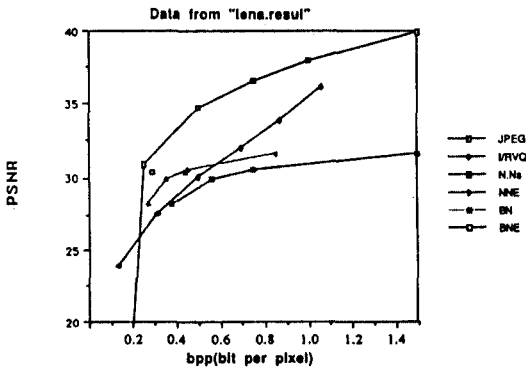


그림 8. Lena 영상을 이용한 결과
Fig. 8. Results for Lena image.

그림 8에서 보듯이 신경회로망만(N.Ns)을 이용한 경우 JPEG과 VQ에 비해 성능이 많이 저하됨을 볼 수 있다. 여기에 entropy coding을 이용한 NNE는 약 0.6 bpp(bit per pixel)이하에서는 I/RVQ보다 성능이 좋아짐을 볼 수 있다. 블럭을 나누는 방법(BN)은 NNE와 비슷한 성능을 나타내므로 블럭을 나누는 것이 유용함을 알 수 있다. 그러나 지금까지 그림 8에 보여준 제안된 영상압축구조가 대표적인 VQ보다 성능이 좋게 나타났으나 아직 JPEG보다는 성능이 떨어진다. 즉 같은 압축률에서 PSNR이 VQ보다는 높고 JPEG보다는 낮다. 이의 가장 큰 이유는 그림 8에서 보듯이 신경회로망을 이용한 N.Ns의 경우 성능이 별로 안 좋으나 압축

률에 대한 화질의 저하가 급격하지 않기 때문이다.

앞의 방법에서 신경회로망 은닉층의 출력을 다시 코딩하는 것으로 블럭분류와 entropy coding을 같이 쓴 BNE는 거의 JPEG과 비슷한 성능을 보였다.

V. 결 론

신경회로망을 이용한 영상압축 및 복원 방법을 제안하여, 이를 바탕으로 블럭 분류기를 이용한 다중구조 신경회로망의 영상데이터 압축과 일반화 특성을 시뮬레이션을 통해 보였다. 특히 2층 신경회로망을 사용하는 영상데이터의 압축은 학습과 계산이 병렬처리된다는 장점을 가지며, 구조가 간단하여 하드웨어 구현에도 용이하다.

시뮬레이션에서 전처리를 하지 않고 신경회로망을 이용한 경우와 클래스로 나누어 압축하는 것을 보였으며, 여기서 entropy coding을 이용하여 약 25:1의 압축률을 얻었다. 이를 JPEG과 VQ와 비교하여 높은 압축률에서 VQ보다 성능이 좋고, JPEG과 비슷한 좋은 성능을 나타냄을 알 수 있었다.

앞으로 할 일은 신경회로망 모델 및 구조를 바꾸어 보는 것과 제안된 구조를 바탕으로 칼라 영상압축, 오디오 코딩에 적용해 보는 것이다.

참 고 문 헌

- [1] "JPEG technical specification : Revision (DRAFT), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVII," August 1986.
- [2] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Kluwer, Norwell, Ma, 1992.
- [3] C. H. Park, "Neuro-fuzzy information processing," Proc. of 91' Korean Scientist and Engineer Summer Symposium, pp. 90-94, Seoul, Korea, 1991.
- [4] N. Sonehara, M. Kawato, S. Miyake, and K. Nakane, "Image Data Compression Using a Neural network Model," IJCNN, vol.2, pp.35-40, Washington, 1989.
- [5] S. Marsi, G. Ramponi and G. L. Sicuranza, "Improved Neural structures for Image

- Compression," ICASSP vol.4, 2821-2824, Toronto, 1991.
- [6] F. Arduini, S. Fioravanti and D. Giusto, "Adaptive Image Coding using Multilayer neural Networks," ICASSP, vol.2, pp.381-384, San Francisco, 1992.
- [7] K. B. Cho, C. H. Park, and S. Y. Lee, "Classified Image Compression and Coding using Neural Networks," Proc. of APCC, pp.414-417, Taejon, Korea, 1993.
- [8] A. Namphol, M. Arozullah, and S. Chin, "Higher order data compression with Neural Networks," IJCNN, vol.1, pp.55-59, Seattle, 1991.
- [9] J. Liu and D. Wang, "Data Compression for Image Recognition using neural networks," IJCNN, vol.4, pp.333-338, Baltimore, 1992.
- [10] L. Russo and E. Real, "Image Compression Using an Outer Product Neural Networks," ICASSP, vol.2, pp.377-380, San Francisco, 1992.
- [11] T. D. Gedeon and D. Harris, "Progressive Image Compression," IJCNN, vol.4, pp.403-407, Baltimore, 1992.
- [12] O. Chen, B. Sheu and W. Fang, "Adaptive Vector Quantizer for Image Compression using Self Organization Approach," ICASSP, vol.2, pp.385-388, San Francisco, 1992.
- [13] R. Lancini and F. Perego, "Neural Networks Approach for Adaptive Vector Quantization of Image," ICASSP, vol.2, pp.389-392, San Francisco, 1992.
- [14] R. P. Lippman, "An Introduction to Computing with Neural Nets," IEEE ASSP magazine pp.4-22, April 1987.
- [15] B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images," IEEE Trans. Commun., vol.34, Nov.1986.
- [16] J. S. Lim, Two-Dimensional Signal and Image Processing, Prentice Hall, USA, 1990.
- [17] M. Rabbani and P. W. Jones, Digital Compression Techniques, SPIE, Washington, 1991.



趙廣輔 (Kwang Bo Cho) 정회원
1966년 12월 4일생
1992년 2월 : 서강대학교 전자공학과 졸업(공학사)
1994년 2월 : 한국과학기술원 및 전자공학과 졸업(석사)
현재 : 삼성중앙연구소 근무중



朴哲勳 (Cheol Hoon Park) 정회원
1961년 4월 14일생
1984년 2월 : 서울대학교 전자공학과 학사
1985년 6월 : California Institute of Technology(석사)
1990년 6월 : California Institute of Technology(박사)
1991년 3월 ~ 1992년 4월 : KAIST태우교수 겸 연구연수원
1992년 5월 ~ 현재 : KAIST 전기 및 전자공학과 조교수

李水榮(Soo Young Lee) 정회원

1952년 10월 15일생

1975년 : 서울대학교 전자공학과 졸업(학사)

1977년 : 한국과학원 전기 및 전자공학과 졸업(석사)

1984년 : Polytechnic Institute of New York 졸업(박사)

1977년 ~ 1980년 : 대한엔지니어링(주) 과장대리

1982년 ~ 1985년 : General Physics Corp. Staff/senior
Scientist

1986년 ~ 현재 : 한국과학기술원 전기 및 전자공학과 조교
수/부교수/교수

※주관심분야 : 신경회로망 이론, 응용(음성인식 및 적응제어), 구현(VLSI 및 광구현)