

연상메모리를 이용한 저해상도 및 저대비 문자 영상 인식

정찬호[○], 김대철[†], 김경환

서강대학교 전자공학과, LG전자[†]

97peter@sogang.ac.kr[○], kimdc@lge.com[†], gkim@sogang.ac.kr

Recognition of character images with low-resolution and low-contrast using an associative memory

Chanho Jung[○], Daechul Kim[†], Gyeounghwan Kim

Dept. of Electronic Engineering, Sogang University

LG Electronics Inc.[†]

요 약

본 논문에서는 저해상도 및 저대비의 특성을 지니는 문자 영상으로부터 특징을 추출하고 연상메모리를 이용하여 대상 문자를 인식하는 방법을 소개한다. 저해상도 영상의 이진화 과정에서 발생할 수 있는 정보의 왜곡 현상을 피하기 위하여 입력 영상의 gradient 정보를 이용하여 특징을 추출한다. 저해상도 및 저대비의 특성을 지니는 문자 영상의 경우 입력 영상에 noise가 존재하거나 충분한 정보가 포함되어 있지 않은 경우 특징벡터에 상당한 왜곡을 초래하게 된다. 손상된 특징을 복원하기 위하여 연상메모리를 이용한다. 인식하고자 하는 문자 영상들의 prototype 영상들을 이용하여 연상메모리의 weight matrix를 구성한다. weight matrix를 이용해서 입력 영상이 가지는 특징과 가장 비슷한 특징을 가지는 prototype 영상의 특징벡터를 생성함으로써 손상된 특징을 복원하게 된다. 제안된 시스템을 이용하여 실험한 결과 noise가 존재하거나 정보가 충분하지 않은 입력 영상에 대해서 비교적 높은 인식률을 얻을 수 있었다.

1. 서 론

인식대상 문자를 구성하는 pixel의 개수가 비교적 적고 그 문자와 배경의 intensity 차이가 크지 않은 경우 문자의 인식을 위한 특징 추출 및 인식 방법의 구현이 쉽지 않다. 특히 입력 영상에 대해 이진화 과정을 수행한 후 특징을 추출할 경우 pixel의 intensity 분포에 의존하는 이진화 과정의 속성으로 인해 noise가 적게 존재할 때에도 특징 값에는 상당한 영향을 끼치게 된다. 따라서 저해상도 및 저대비 문자 영상의 경우 이진화 과정을 거치지 않은 입력 영상 자체에서의 특징 추출이 필요하다[1].

본 논문에서는 저해상도 및 저대비의 특성을 지니는 문자 영상에서 gradient 정보를 이용하여 특징을 추출하고 입력 영상 자체의 왜곡에 의해 손상된 특징의 복원을 위해 연상메모리를 이용하는 알고리즘을 제안한다. Gradient 정보를 이용하여 특징을 추출하는 과정에서는 gradient의 크기 성분에 대해서 adaptive thresholding 과정을 수행하고 gradient의 크기가 비교적 큰 부분에 대해서 gradient의 방향 성분을 이용하여 특징 벡터를 구한다. 손상된 특징의 복원은 인식하고자 하는 문자 영상들의 prototype 영상들의 특징 벡터를 이용하여 연상메모리의 weight matrix를 구하고 그를 이용함으로써 수행된다.

2. Gradient 정보를 이용한 특징 추출

그림 1은 특징벡터를 생성하기 위한 전처리 과정이다. 일반

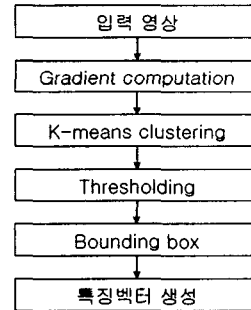


그림 1. 특징벡터를 생성하기 위한 전처리 과정

적으로 적용되는 이진화 과정과 그를 기반으로 하는 특징 추출 과정에서 있을 수 있는 정보 손실을 최소화하기 위해 gradient 정보를 이용하여 Sobel mask를 이용해서 구한다. 특징 추출 과정에서 gradient의 크기는 이중 가우시안 분포를 갖는다고 가정하고 gradient의 크기에 대해서 K-means clustering을 수행한다. K-means clustering을 수행하면 gradient의 크기가 상대적으로 큰 부분과 작은 부분으로 나눌 수 있게 된다. 상대적으로 큰 부분의 평균과 분산을 각각 μ_1, σ_1 이라 하고 작은 부분의 평균과 분산을 각각 μ_2, σ_2 라 할 때 $\mu_2 - 2\sigma_2$ 를 threshold 값으로 하여 gradient의 크기에 대하여 thresholding 과정을 수행한다. $\mu_2 - 2\sigma_2$ 의 값은 문자와 배경의 intensity 차에 따라서 다른 값을 가지므로 영상에 adaptive한 threshold 값이

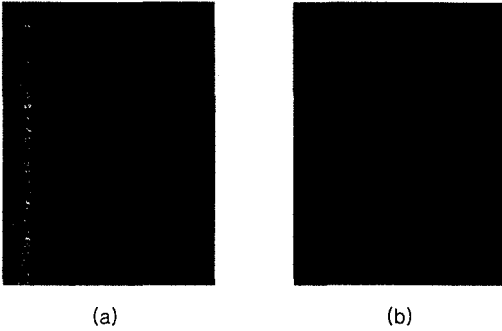


그림 2. Gradient 크기의 thresholding 과정 : (a) Thresholding 과정을 수행하기 전의 gradient 영상, (b) Thresholding 과정을 수행한 후의 gradient 영상

라 할 수 있다. 그림 2는 gradient 크기의 thresholding 과정을 보여준다. 그림 2를 보면 gradient의 크기와 방향이 화살표의 크기와 방향으로 나타나 있음을 볼 수 있다. 입력 영상에서 문자와 배경의 intensity 차이가 크지 않음에도 불구하고 문자의 contour 부분에서는 gradient의 크기가 비교적 큰 값을 나타낸다. 그리고 gradient의 방향 성분은 edge의 방향 성분과는 약 $\pm 90^\circ$ 의 관계를 보임을 알 수 있다. 특징벡터는 gradient의 방향 성분을 이용해서 얻어진다. Thresholding 과정을 거친 gradient 영상에 bounding box를 적용하고 전체 영상은 3×3 region으로 나눈다. 나누어진 각 region에 대해서 각 방향 성분에 대한 region내에서의 비율을 구한다. 특징벡터는 식 1과 같이 표현된다.

$$F_i(k) = \alpha_k \quad (1 \leq i \leq 9, 1 \leq k \leq 9) \quad (1)$$

$$\begin{cases} k = \lfloor \frac{\theta_k}{20^\circ} \rfloor + 1 & (0^\circ \leq k < 180^\circ) \\ k = \lfloor \frac{\theta_k - 180^\circ}{20^\circ} \rfloor + 1 & (180^\circ \leq k < 360^\circ) \end{cases}$$

식 1에서 i 는 각 region의 index를 나타내고 θ_k 는 각 region 내에서 gradient의 방향 성분의 범위를 나타낸다. 임의의 edge 방향 성분이 존재할 때 약 $\pm 90^\circ$ 의 차이를 보이는 gradient의 방향 성분이 존재하므로 약 90° 차이를 보이는 gradient의 방향 성분과 약 -90° 차이를 보이는 gradient의 방향 성분을 동일한 방향으로 가정하고 각 region 내에서 gradient의 방향 성분의 비율을 구한다. 그림 3은 gradient 방향 성분의 quantization 과정을 보여준다. 따라서 α_k 는 각 region 내에서 gradient의 방향 성분들이 k 번째 범위에 속하는 비율이다. 따라서 식 2가 성립한다.

$$0 \leq \alpha_k \leq 1 \quad (2)$$

식 1, 2에서 보는 바와 같이 특징벡터의 dimension은 81이고 특징벡터의 각 component는 0과 1사이의 값을 갖는다.

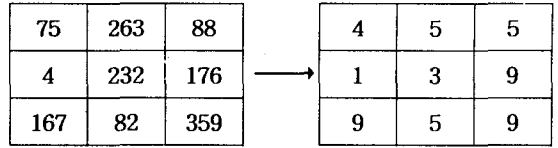


그림 3. Gradient 방향 값의 quantization 과정

3. 연상메모리를 이용한 손실된 특징의 복원

연상메모리는 출력되기를 원하는 pattern들을 메모리에 저장함으로써 동작한다. 임의의 입력 pattern에 대하여 메모리에 저장되어 있는 pattern들과 비교하여 가장 유사한 pattern을 출력한다. 따라서 입력 data의 일부뿐만 연상메모리에 입력되어도 연상메모리의 출력에서는 입력 data와 가장 유사한 data를 출력하게 된다. 그림 4는 연상메모리의 동작을 간단하게 보여준다.

본 논문에서는 입·출력 unit의 개수가 같은 auto associative memory 구조이면서 feed-forward network 형태를 지니는 linear associative memory 구조를 갖는 연상메모리를 사용한다. 또한 입·출력 unit을 구성하는 값이 discrete value인 discrete associative memory 구조를 갖는 연상메모리를 사용한다. 연상메모리의 weight matrix는 식 3의 과정에 의해서 구해진다.

$$W = \sum_{i=1}^n P_i P_i^T \quad (3)$$

식 3에서 P_i 는 출력되기를 원하는 각각의 pattern들의 column 벡터를 나타내고 T 는 matrix의 transpose를 의미한다. 또한 n 은 출력되기를 원하는 pattern의 수를 나타낸다. 따라서 각각의 pattern들의 column 벡터의 dimension이 n 이라면 weight matrix W 의 dimension은 $n \times n$ 이 된다.

본 논문에서는 gradient 정보를 이용하여 얻은 특징벡터를 출력되기를 원하는 pattern들로 보고 weight matrix를 구성한다. 또한 각각의 특징벡터들을 구성하는 component들이 discrete한 값을 가지므로 각각의 component들을 0과 1로 구성하기 위하여 임의의 threshold 값 τ 를 이용하여 특징벡터를 재구성한다. 특징벡터의 재구성은 식 1에서 얻어진 특징벡터들 식 4와 같이 thresholding 함으로써 이루어진다.

$$F_i(k) = \begin{cases} 0 & (\text{if } \alpha_k < \tau) \\ 1 & (\text{otherwise}) \end{cases} \quad (4)$$



그림 4. 연상메모리의 동작 예

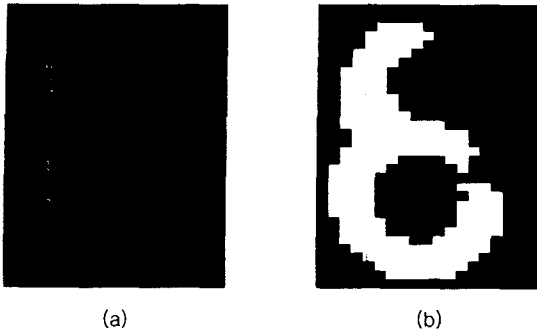


그림 5. 입력 영상의 특징 추출만으로는 정확한 인식 결과를 얻기 어려운 영상의 예 : (a) 입력 영상, (b) 입력 영상의 이진화 결과

따라서 인식하고자 하는 문자 영상의 prototype 영상들의 특징벡터를 생성하여 weight matrix를 구성한다. Prototype 영상들의 특징벡터를 생성할 때는 식 4를 이용하여 특징벡터의 component들이 0과 1의 값을 가지도록 한다.

그림 5는 입력 영상을 이진화 한 후 특징을 추출하거나 입력 영상 자체에서 특징을 추출해서 인식할 경우 정확한 인식 결과를 얻기 어려운 영상의 예를 보여준다. 그림 5와 같이 불균형한 조명 조건이나 입력 영상 자체의 왜곡에 의해 특징벡터가 크게 손상된 경우 연상메모리를 이용해서 손상된 특징벡터를 복원해 낼 수 있다. 손상된 특징벡터를 복원해 내는 과정은 식 5와 같은 과정을 통해서 이루어진다.

$$Y = WX \tag{5}$$

식 5에서 X 는 연상메모리의 입력 벡터를 의미하고 Y 는 연상메모리의 출력 벡터를 의미한다. 그리고 W 는 weight matrix를 의미한다. 입력 영상에 noise가 존재하거나 충분한 정보가 포함되어 있지 않은 경우 입력 벡터 X 에 심한 왜곡이 발생하는데 식 5와 같은 과정을 거치게 되면 연상메모리에 저장되어 있는 prototype 영상들의 특징벡터와 입력 벡터를 비교하여 입력벡터와 가장 비슷한 prototype 영상의 특징벡터를 출력하게 된다. 그러나 prototype 영상들의 특징벡터들이 서로 orthogonal하지 않을 경우 정확하게 prototype 영상의 특징벡터를 출력하지 못하므로 식 4의 threshold 값 τ 를 적절히 조정하여 prototype 영상들의 특징벡터간의 hamming distance를 크게 만들어야 한다.

4. 실험 결과 및 분석

본 논문에서 사용한 실험 영상은 약 20×30, 10×20의 크기를 가지는 저해상도 및 저대비의 특성을 지니는 0~9의 숫자 영상이다. 전체 1640장의 영상들에 대해서 비교적 noise가 적

표 1. 제안한 알고리즘을 이용한 실험 결과와 비교 실험 결과 : (a) Gabor filter를 이용한 특징 추출 방법을 이용한 인식기의 인식률, (b) Concavity 특징 추출 방법을 이용한 인식기의 인식률, (c) Gradient 정보를 이용한 특징 추출 방법을 이용한 인식기의 인식률, (d) Gradient 정보를 이용한 특징 추출 방법과 concavity 특징 추출 방법을 함께 이용하고 연상메모리를 이용한 인식기의 인식률

	class A(1245장)	class B(395장)
(a)	79.76%	56.58%
(b)	94.93%	58.75%
(c)	96.39%	74.95%
(d)	96.31%	80.54%

고 gradient가 잘 표현된 영상들(class A)과 불균형적인 조명이나 입력 영상 자체의 왜곡에 의한 noise가 포함된 영상들(class B)로 나누어 실험을 진행하였다. 또한 gradient 정보를 이용한 특징 추출과 입력 영상의 이진화를 통한 특징 추출을 비교하기 위하여 gabor filter를 이용한 특징 추출 방법[2]과 concavity 특징 추출 방법[3]을 비교하여 실험하였다. 또한 prototype 영상들의 특징벡터들이 서로 orthogonal하지 않은 것을 보완해 주기 위하여 gradient 정보와 concavity 정보를 함께 이용하고 연상메모리를 이용하여 실험하였다. 이 때 특징벡터의 dimension은 111이 되고 그 중 81개의 component가 gradient 정보를 이용하여 구해진다. 표 1을 보면 입력 영상의 이진화를 통한 특징 추출 방법을 이용하는 것보다 gradient 정보를 이용한 특징 추출 방법을 이용한 인식기의 성능이 우수함을 알 수 있고 연상메모리를 사용하면 비교적 noise가 적고 gradient가 잘 표현된 영상들뿐만 아니라 불균형적인 조명 조건이나 입력 영상 자체의 왜곡에 의한 noise가 포함된 영상들에 대해서도 비교적 높은 인식률을 보임을 알 수 있다.

5. 참고 문헌

[1] G. Shrikantan, S. Lam, and S. Srihari, "Gradient-based contour encoding for character recognition," *Pattern Recognition*, vol. 7, no. 29, pp. 1147-1160, 1996.
 [2] P. Hu, Y. Zhao, Z. Yang, and J. Wang, "Recognition of gray character using gabor filters," *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, pp. 419-424, 2002.
 [3] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic recognition of handwritten numerical strings: A recognition and verification strategy," *IEEE Trans. on PAMI*, vol. 24, no.11, pp. 1438-1454, 2002.