

논문 2004-41CI-2-7

## NMF를 이용한 영문자 활자체 폰트 분류

### (Font Classification of English Printed Character using Non-negative Matrix Factorization)

이 창 우\*, 강 현\*, 정 기 철\*\*, 김 향 준\*

(Chang-Woo Lee, Hyun-Kang, Kee-chul Jung, and Hang-Joon Kim)

#### 요 약

최근 대부분의 문서들이 전자적으로 생성되고 많은 고문서(古文書)들이 이미지 형태로 전자화되고 있다. 이미지 형태의 전자 문서들은 정보 추출과 데이터베이스화에 많은 어려움이 있기 때문에, 이러한 문서를 효율적으로 관리하고 검색하기 위한 문서구조분석 방법과 문자 인식을 위한 많은 연구가 필요하다. 본 논문은 폰트의 구분 특성(font discrimination features)들이 폰트이미지의 공간적으로 지역적인 특징들에 기반함을 가정한 방법으로, 객체의 부분기반 표현들을 학습할 수 있는 NMF(non-negative matrix factorization) 알고리즘을 사용하여 폰트를 자동으로 분류하는 방법이다. 제안된 방법은 부분기반의 비지도 학습 방법(part-based unsupervised learning technique)을 이용하여 전체의 폰트 이미지들로부터 각 폰트들의 구분 특징인 부분을 학습하고, 학습된 부분들을 특징으로 사용하여 폰트를 분류하는 방법이다. 실험결과에서 폰트 이미지들의 공간적으로 국부적인 특징들이 조사되고, 그 특징들이 폰트의 식별을 위한 적절성을 보인다. 제안된 방법이 기존의 문자인식, 문서 검색 시스템들의 전처리기로 사용되면, 그 시스템들의 성능을 향상시킬 것으로 기대된다.

#### Abstract

Today, most documents are electronically produced and their paleography is digitalized by imaging, resulting in a tremendous number of electronic documents in the shape of images. Therefore, to process these document images, many methods of document structure analysis and recognition have already been proposed, including font classification. Accordingly, the current paper proposes a font classification method for document images that uses non-negative matrix factorization (NMF), which is able to learn part-based representations of objects. In the proposed method, spatially local features of font images are automatically extracted using NMF, then the appropriateness of the features specifying each font is investigated. The proposed method is expected to improve the performance of optical character recognition (OCR), document indexing, and retrieval systems, when such systems adopt a font classifier as a preprocessor.

**Keywords :** Font classification, NMF, Part-based unsupervised learning, OCR

#### I. 서 론

기존 OCR 시스템에 폰트 식별기(font recognizer)가 전처리기로 사용될 경우, 그 성능이나, 정확도 및 처리 시간 면에서 상당한 개선을 기대할 수 있다. 또한, 대부분의 문자 인식 시스템에서 폰트 식별은 다양한 글자들의 변형을 처리해야 하는 부담을 줄여주는 장점이 있다 [1].

폰트 식별이 문서 이미지 처리나 인식 시스템에서 필수 불가결한 요소임에도 불구하고, 아직 많은 시스템들이 이를 채택하지 않고 있으며, 최근에는 폰트 식별에

\* 정회원, 경북대학교 컴퓨터공학과  
(Dept. of Computer Eng. at Kyungpook National Univ.)

\*\* 정회원, 숭실대학교 미디어학부 - 교신저자  
(College of Information Science at Soongsil Univ.)  
※ 이 논문은 한국과학재단의 해외 Post-doc. 연수지원에 의하여 연구되었음  
접수일자: 2003년9월8일, 수정완료일: 2004년2월15일

대해 많은 국내외 연구가 보고 되고 있다<sup>[1-5]</sup>. 이전 폰트 분류 방법들은 문서 이미지 전체적인 특징(global property)<sup>[2,3,5]</sup>과 국부적인 특징(local property)를 사용하는 방법<sup>[4,13]</sup>으로 구분할 수 있고, 글자 단위의 분류 방법<sup>[13]</sup>과 단어나 문장의 길이 단위의 분류<sup>[2]</sup>, 블록단위의 분류 방법<sup>[5]</sup>들로 크게 구분할 수 있다. 그 예로 Khoubyari와 Hull은 한 장의 문서이미지에서 대표적인 폰트와 흔히 사용되는 기능성 단어(functional words)들을 식별하기 위한 방법을 제안하였다<sup>[2]</sup>. Khoubyari와 Hull의 연구에서는 먼저 입력 문서로부터 단어 이미지들의 클러스터를 생성하여 데이터베이스화 하고, 실험단계에서는 문서 이미지들로부터 기능성 단어들을 추출한 다음, 데이터베이스화 되어진 클러스터의 템플릿들과 비교하여 폰트를 분류하였다. 이와 같은 방법은 'a'나 'the'와 같은 기능성 단어들의 데이터베이스를 미리 구성해야 하는 부담(overhead)이 존재한다. Shi와 Pavlidis는 문자 인식의 정확성을 높이기 위해 문맥 정보(contextual information)와 폰트 인식 결과를 결합한 방법을 제안하였다<sup>[3]</sup>. 이들은 폰트의 인식을 위하여, 전체적인 페이지의 성질인 획의 기울기와 특정 길이에 대한 단어의 히스토그램, 그리고 인식된 짧은 단어들에 대한 매칭 결과를 폰트의 특징 정보로 추출하였다. 이러한 방법은 serif와 sans-serif 폰트들 그리고, upright와 slanted 폰트들로 분류하는 것이 목적이었으며, Arial, Courier, Gothic와 같은 활자체에 초점을 맞추진 못했다. Zrandini와 Ingold의 논문에서는, 분류하고자 하는 텍스트의 어떤 문맥에 대한 정보도 없이 이미지 블록들로부터 활자체, 굵기, 기울기, 문자 크기를 식별하기 위해 전역적인 인쇄상의 특징들을 기반으로한 통계적 방법을 사용하였다<sup>[4]</sup>. 그러나 한 폰트를 구별하기 위한 노력으로 폰트 특징 추출을 위한 세밀한 노력이 필요하고, 또한 일정 길이 이상의 텍스트 길이를 요구한다. 반면에 Zhu et al는 각 폰트의 상세한 지역적 특징을 분석하지 않고, 폰트 인식을 위해 텍스처 분석 기반 방법을(texture analysis based approach) 제안하였다<sup>[5]</sup>. 이 방법은 문서이미지에서 공백이 없는 글자 블록을 형성하고, 이 블록으로부터 폰트를 대표하는 텍스처를 추출한다. 이 경우, 폰트 식별은 문자(character) 또는 단어 레벨의 분류가 아니라 이미지의 블록단위로 수행되어진다. 이 방법은 텍스트들로 구성된 블록을 구성해야 하는 부담이 있고, 한 블록 내에 포함된 여러 다른 폰트를 구별할 수 없는 단점이 있다

본 논문에서는 NMF를 사용한 폰트 식별 방법을 제안한다. 제안한 방법은 이미지 공간에서 각각의 폰트들이 차지하고 있는 공간적 지역성(spatial locality)이 다르다는 것을 폰트를 분류하는 특징으로 간주한 방법으로써, 객체의 부분기반 표현들을 학습할 수 있는 NMF(non-negative matrix factorization) 알고리즘을 사용하여 폰트 이미지를 부분별로 분리하여 학습한다. 제안된 폰트 분류 방법은 폰트 이미지들의 집합으로부터 각 폰트를 구별하기 위한 지역적 특징들을 자동적으로 추출해 주는 NMF 알고리즘을 사용한다(그림 1).

학습 단계에서, 열벡터(column vectors)들의 집합인 폰트 이미지들의 집합이 주어지면, NMF는 주어진 폰트의 집합을 엔코딩(encoding,  $H$ )과 베이스(basis,  $W$ )로 분해(factorization)한다. 분해의 과정 동안, 베이스는 폰트 이미지들로부터 지역적 특징들을 학습하고, 엔코딩들은 각 폰트를 대표하는 템플릿을 구성한다. 테스트 단계에서 학습 단계에서 학습된 베이스를 사용하여, 문서 이미지로부터 분리된 문자 이미지 하나가 그 이미지에 상응하는 하나의 엔코딩을 생성한다(본 논문에서는 이 과정을 NMF 공간으로의 투영이라 칭한다). 그리고 NMF 공간으로 투영된 엔코딩이 템플릿 폰트들과 시티블럭 거리(City-Block distance 혹은, L1 distance)를 비교하여 가장 가까운 거리에 있는 폰트로 식별된다(본 논문에서 이 과정을 NNC(nearest neighbor classifier)라 칭한다). 결과적으로 NMF는 폰트 이미지들의 세트로부터 각 폰트들의 특징을 결정짓는 국부적인 특징들을 자동적으로 추출할 수 있고, 이를 폰트 분류를 위한 특징으로 사용한다. 제안된 폰트 분류 방법은 문자 단위로 분류 결과를 알 수 있기 때문에, 문자 인식 시스템에 응용하기 위해 단어 또는 블록 레벨의 폰트 분류 방법들보다 더 적합한 방법이다.

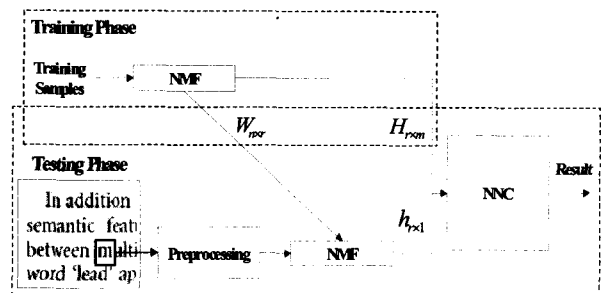


그림 1. 제안된 폰트 분류 방법의 흐름도.  
Fig. 1. Flow diagram of proposed font classifier.

본 논문에서는 다음과 같은 순서에 따라 제안된 방법을 설명한다. II장에서는 NMF 기술에 대한 개괄을 설명하고, III장에서는 NMF를 사용한 폰트 분류 방법에 대해 기술한다. IV장에서는 실험 결과를 보이고, V장에서는 결론과 향후과제에 대해 기술한다.

## II. NMF 알고리즘

부분기반의 비지도 학습이란 비지도 학습의 일종으로 객체의 부분을 학습할 수 있는 학습 방법이라 할 수 있다. 이와 같은 맥락으로 컴퓨터 비전 분야에서 객체의 표현을 위해 국부적인 특징을 추출하려는 몇몇 시도가 있었다<sup>[14-16]</sup>. 예를 들어 Weber는 한 객체 클래스는 공간적, 시각적으로 유사한 특징들을 공유하는 객체들의 집합이다라는 가정을 사용하였다<sup>[16]</sup>. 이 방법은 하나의 객체가 다른 여러 부분들로 구성되어있다고 보는 것이고, 모든 부분들의 조합이 그 객체 전체를 표현한다는 것이다. 그래서 Weber는 국부특징 검출기(local feature detector)를 사용하여 객체의 특징을 추출하였고, 추출된 특징들로부터 객체를 표현하기위해 클러스터링 알고리즘을 적용하여, 가장 두드러진 특징들을 선별하였다. 또한, Bierderman은 Recognition By Components(RBC) 모델을 소개하였고, geons이라 불리는 20여 개의 고유 부분 모양(generic shape parts)을 제안하였다<sup>[14]</sup>. 또한 Bierderman은 구조적 분해 (structural decomposition)만이 객체 처리 (object processing)의 모든 문제를 다룰 수 있다고 주장하였다. Bierderman의 연구를 바탕으로 Shams는 geons에 대한 어떠한 사전지식도 없이 객체의 전체를 구성하는 각각의 geons을 학습하는 방법을 제안하였다<sup>[15]</sup>. 앞서 언급한 다양한 연구와 같이, 한 객체의 전체적인 표현은 그 객체를 구성하고 있는 부분들의 조합으로 정의하고, 이 부분들에 대한 사전지식 없이 그 부분들을 학습하는 것을 부분기반의 비지도 학습이라 할 수 있다.

최근 Lee와 Seung에 의해 발표된 NMF 알고리즘은 주어진 행렬을 베이스스 행렬과 엔코딩 행렬로 분해하는 방법이다<sup>[6]</sup>. 이 알고리즘이 주성분분석(Principal Component Analysis, PCA) 방법이나 벡터양자화(Vector Quantization, VQ) 방법과 같은 행렬분해 알고리즘들과 가장 큰 차이는 분해된 각 행렬의 요소들이 모두 0 이상의 값들이라는 것이다. NMF 알고리즘은 수식(1)에서처럼  $n \times m$ 의 행렬  $V$ 를  $n$ 차원의 비음수 벡터  $r$ 개로 구

성된 베이스스( $W$ )와  $r$ 차원의 비음수 벡터  $m$ 개로 구성된 엔코딩( $H$ )으로 분해한다. 여기서 랭크(rank)  $r$ 은 일반적으로  $(n+m)r < nm$ 의 범위 내의 임의의 수로 결정한다.

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{a=1}^r W_{ia} H_{a\mu} \quad (1)$$

NMF 알고리즘은  $W$ 와  $H$  행렬의 랜덤 초기화로부터 시작하여, 두 행렬의 곱이 원 행렬에 수렴될 때까지 반복적으로 갱신(update)한다. 알고리즘의 수렴 정도를 측정하기 위해 평가 함수(objective function)가 필요하고, 이 평가 함수는  $H$ 와  $W$ 행렬의 곱이 원 행렬  $V$ 를 생성하기위한 유사도(likelihood)와 연관된다. NMF 알고리즘에서는 원 행렬  $V$ 는 평균이  $WH$ 인 포아송 분포(Poisson distribution)로부터 유도된다고 가정한다. 따라서 포아송 분포는 수식(3)처럼 표현된다<sup>[7]</sup>. 수식(3)에서 양변에 로그를 취한 수식(4)는  $V$ 에만 연관된 함수이기 때문에  $\log V!$  항은  $W$ 와  $H$ 에 관하여 최적화 시에 별다른 영향을 주지 않기 때문에 제외되어도 무방한 항이다. 이와 같은 방식으로 수식(2)와 같은 평가 함수가 유도된다.

$$F = \sum_{i=1}^n \sum_{\mu=1}^m V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu} \quad (2)$$

$$P(V | WH) = \exp(-WH) \frac{(WH)^V}{V!} \quad (3)$$

$$\log P(V | WH) = V \log(WH) - WH - \log V! \quad (4)$$

NMF 알고리즘은 반복적 알고리즘으로 곱셈만으로 이루어진 갱신 규칙(update rule)을 가지고 있다. 특히 이 알고리즘은 gradient-descent 알고리즘의 변형으로 간주될 수 있다<sup>[8]</sup>. NMF 알고리즘은 수식(5)에서와 같이 곱셈 갱신 규칙에 따라  $W$ 와  $H$ 를 반복적으로 갱신한다.

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}}, \quad W_{ia} \leftarrow W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}},$$

$$W_{ia} \leftarrow W_{ia} \frac{W_{ia}}{\sum_j W_{ja}} \quad (5)$$

베이스스  $W$ 가 특정 데이터 클래스를 표현하기위한 특징들을 학습한 후에는, 수식(6)에서와 같이 베이스스 행렬  $W$ 는 그대로 사용하여 새로운 테스트 벡터  $v$ 를

NMF 공간으로 사상시킨다. 이때, 임의로 초기화 되어진  $H$  행렬만을 갱신함으로써 테스트 벡터  $v$ 에 상응하는 엔코딩  $h$ 를 구할 수 있다. 역으로, 입력영상은 [그림 2]에서 보여 지는 것처럼 수식(6)을 사용해서 재구성되어진다. 이러한 방식으로 입력 영상에 상응하는 새로운 엔코딩  $h$ 를 구할 수 있고, 결과적으로 베이스  $W$ 는 갱신 없이 사용되면서 새로운 엔코딩  $h$ 를 계산할 수 있다. 즉, 예를 들면 한 장의 이미지 벡터  $v(n \times 1)$ 는 학습단계에서 생성된  $W$  행렬( $n \times r$ )을 그대로 사용하면서  $h(r \times 1)$  행렬만을 수식(5)의 갱신규칙을 이용하여 수식(2)의 평가함수가 최소가 되도록 갱신하여 구한다.

$$v_{n \times 1} \approx (W_{n \times r})h_{r \times 1} = \sum_{a=1}^r \sum_{i=1}^n W_{ia} h_{a1} \quad (6)$$

[그림 2]에서는 6개의 다른 폰트에 대한 4992개 문자 이미지들을 대상으로 구한 베이스 이미지들과 엔코딩을 보인다. 각각의 폰트 이미지들의 크기는  $28 \times 28$  픽셀이고, 모든 픽셀에 대한 밝기값은  $[0,1]$  범위로 정규화하여 사용한다. [그림 2]에서 보는 것처럼, NMF의 베이스와 엔코딩은 많은 개수의 구성 요소들이 0에 가까운 값이거나, 혹은 0이기 때문에, 베이스 이미지와 엔코딩 이미지가 희소행렬(sparse matrix)의 형태를 띄고 있음을 쉽게 알 수 있다. 다시 말해서, 주어진 문자 이미지는 사용할 수 있는 베이스 전부를 사용하지 않고, 그 문자를 표현하기 위해 일부의 베이스만을 사용하기 때문에 한 이미지에 상응하는 엔코딩 또한 희소행렬의 형태를 띄고 있다.

[그림 2]에서 주어진 행렬  $V(784 \times 4992)$ 가  $W(784 \times 100)$ 와  $H(100 \times 4992)$  행렬로 분해됨을 도시한 그림이다. 즉, 비음수 행렬 분해 알고리즘의 파라메타인  $n, m, r$ 은 각각 784, 4992, 100이다. 이때 종료조건으로 갱신의 반복횟수를 사용하였고, [그림 2]에서의 결과는 300회 반복한 결과이다. [그림 2]의 오른쪽 상단의 원 문자 이미지는 [그림 2]의 왼쪽에 보이는 베이스의 칼럼들과 중간부분에서 굵은 선으로 표현된 엔코딩의 가중치를 가진 선형 조합으로 표현된다. 엔코딩 이미지에서 굵은 선으로 표현된 엔코딩은 원 문자 이미지와 일대일 대응 관계를 가진다. [그림 2]에서 베이스와 엔코딩을 가시화하기 위해  $[0,255]$  사이로 각 구성 요소들을 정규화하여 표현하였다. 각 베이스와 엔코딩을 표현한 그림에서는 각 구성요소의 값이 높을수록 더 밝은 밝기 값을 가진다.

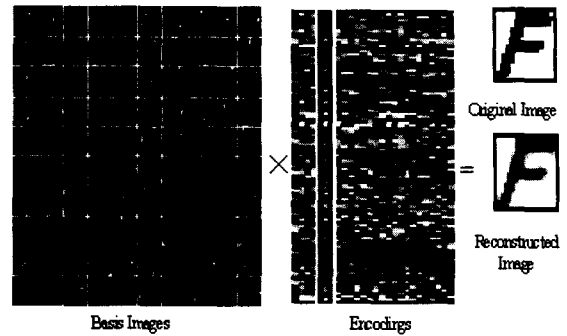


그림 2. NMF 알고리즘 수행의 예.

Fig. 2. Image reconstruction using factorized basis and encoding.

### III. 제안된 폰트 분류 방법

제안된 방법은 이미지 공간에서 각각의 폰트들이 차지하고 있는 공간적 지역성(spatial locality)이 다르다는 것을 폰트를 분류하는 특징으로 간주한 방법으로써, 폰트 이미지 전체의 지역적인 부분들이 특정 폰트를 잘 표현할 것이라는 생각에 기반한 방법이다. NMF는 분해된 행렬  $W$ 와  $H$  내의 어느 구성 요소들도 음수가 아니고, 이러한 비음수 성질을 이용한 제약 조건이, 전체는 각 부분들의 합으로 구성된다는 직관적인 개념의 성립을 가능하게 한다.

주성분분석 방법이나 벡터양자화 방법과 같은 방법들과 비교할 때, 주성분분석 방법에서 구해진 베이스 이미지는 각 베이스의 직교성만을 강조하기 때문에 국부적이거나 희소하지 않을 뿐만 아니라 음의 요소를 허용하기 때문에 인간의 직관을 위배하게 된다. 벡터양자화 방법에서 구해진 베이스는 winner-take-all의 강력한 제약조건의 사용으로 구성 요소에 음의 요소를 허용하지는 않지만, 주성분 분석과 마찬가지로 국부적이거나 요소들이 희소하지 않다<sup>[6]</sup>.

NMF 알고리즘에서 베이스의 희소성의 결과로써, 한 이미지의 엔코딩 또한 [그림 2]에서 보여지는 것처럼, 투영된 공간에서 희소하게 분포된다. 결과적으로, 폰트 이미지들의 지역적 부분들을 학습하는 NMF의 베이스는 각 폰트의 특징들을 표현하고, 한 폰트 이미지가 베이스(폰트이미지의 부분) 이미지의 일부분만을 사용하여 표현된다는 성질을 이용하여 폰트를 분류한다. [그림 1]처럼, 제안된 폰트 분류 방법은 훈련 테스트 이미지들을 이용하여 템플릿 엔코딩을 만들고, 폰트의

분류를 위해 문서 이미지로부터 얻어진 테스트 이미지와 비교하는 방법으로 구성된다. NNC는 아래의 수식(7)과 같이 테스트 이미지로부터 추출한 엔코딩과 템플릿 엔코딩들을 비교하여 가장 가까운 거리에 있는 템플릿의 폰트  $k$ 로 분류한다. 수식 (7)에서  $dist(\cdot)$ 는 시티브릭 거리 척도이며,  $H_k$ 와  $h$ 는  $k$  번째 템플릿 엔코딩과 테스트 이미지의 엔코딩을 의미한다.

$$K = \arg \min_k dist(H_k, h) \quad (7)$$

[그림 3]은 한 테스트 이미지를 NMF 공간으로 사상했을 때, 그 엔코딩과 템플릿 엔코딩들을 비교하여, 최소 거리를 가진 한 템플릿 엔코딩을 도식화 한 예이다. [그림 3]에서 '□'는 테스트 이미지의 엔코딩을 의미하고, '●'는 이 테스트 이미지의 엔코딩과 최소 거리를 가진 템플릿 엔코딩을 나타낸다. 그림 3에서는 주어진 훈련 샘플들로부터 NMF 알고리즘을 수행할 때  $r=100$ 인 경우를 도식화하였다. [그림 3]에서 볼 수 있듯이 하나의 특정 폰트 이미지는 분해된  $W$  행렬을 상수로 사용하여 새로이 NMF 공간으로 사상했을 때, 그 폰트 이미지의 템플릿 엔코딩과 유사한 엔코딩을 생성함을 알 수 있다. 이는 동일한 폰트 이미지는 같은 베이스스 이미지(폰트 이미지의 부분들)를 사용한다는 것을 의미한다. 따라서 제안된 방법은 동일한 폰트 이미지들이 공간적인 국부성을 공유한다는 것에 기반하여 폰트를 분류한다

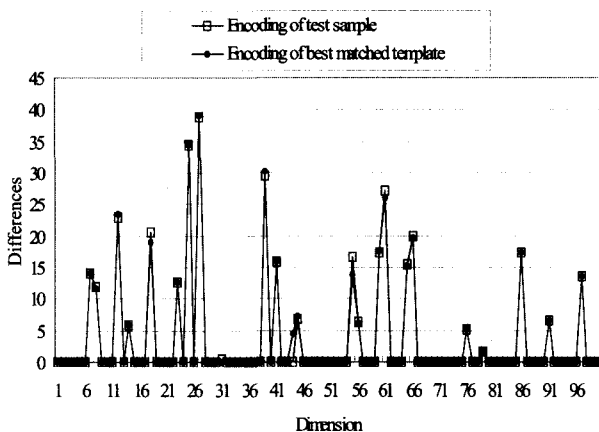


그림 3. Courier capital 'A' 입력 이미지의 분류 예.  
Fig. 3. Example of classification result using Euclidean distance for Courier capital 'A'.

#### IV. 실험 결과

실험에 사용된 폰트는 빈번하게 사용되어지는 영어 폰트 6개 (Courier New (CN), Comic Sans MS (CS), Impact (IM), MS Gothic (MG), Microsoft Sans Serif (MS), Times New Roman (TR))이다. 본 논문에서 분류는 6개의 폰트×4개의 폰트 스타일(굵게, 보통, 기울기, 굵게+기울기)×2개의 형태(대문자, 소문자) 중에 하나로 분류하였다. 즉, 테스트 영상은 48개의 글꼴 유형들 중 하나로 분류된다.

글자 크기를 14 픽셀로 노이즈가 없는 폰트 이미지들과 각각 100, 200 그리고, 300 dpi의 해상도로 스캔된 이진 영상들로부터 획득한 4992개의 문자 이미지들을 훈련 이미지로 채택하였다. 훈련에 사용된 이미지는 모두 28×28 크기의 이미지로 정규화하여 사용하였고, 모든 폰트 이미지를 열벡터로 하나의 전체 행렬을 구성하였다. 이렇게 구성된 전체 행렬을 NMF 알고리즘을 이용하여, 베이스스  $W$ 와 엔코딩  $H$ 로 분해하였다. 테스트 단계에서 엔코딩들은 각 폰트 유형을 표현하는 템플릿 엔코딩들로서 사용되었고, 테스트 테이터는 인쇄된 144개의 문서 이미지를 스캔하여 사용하였다. 즉, 각각 100, 200, 300dpi의 해상도를 가진 이진 영상으로써, 실험 대상이 된 모든 글꼴유형에 대해 워드프로세서를 이용하여 생성, 인쇄, 그리고 스캔하여 사용하였다.

100dpi의 해상도로 스캔된 문서에서는 한 장의 이미지 사이즈가 651×888화소에서 655×920까지의 범위를 가지며, 200 dpi의 해상도 문서에서는 1311×1737화소에서 1287×1904까지의 범위를 가지고, 300 dpi의 해상도 문서에서는 1960×2642화소에서 1960×2809까지의 범위를 가진다. 테스트 이미지의 총 개수는 190830 개의 문자 이미지들이다. 이는 각각 Courier New 폰트 10368×3(3종류의 해상도)개의 문자 이미지, Comic Sans 폰트 8544×3개의 문자 이미지, Impact 폰트 11586×3개의 문자 이미지, MS Gothic 폰트 10416×3개의 문자 이미지, Microsoft Sans Serif 폰트 11464×3개의 문자 이미지, 그리고 Times New Roman 폰트 11232×3개의 문자 이미지로 구성된다. 테스트와 훈련단계에서 모든 문서들에 대해, 글자는 흰색으로 나머지부분은 검정색으로 표현하였다. [표1]은 각 문서 이미지 안에 포함된 글자수를 나타낸다

[그림 4]는 테스트에 사용된 문서 영상들의 일부분이다. [그림 4]는 보통 글자체(regular type)에 대문자들만

표 1. 각 문서 이미지에 포함된 문자의 수.

Table 1. The number of characters included in each document images.

	Regular		Bold		Italic		Bold Italic		Total
	capital	small	capital	small	capital	small	capital	small	
CN	1296	1296	1296	1296	1296	1296	1296	1296	10368
CS	938	1204	933	1197	938	1204	933	1197	8544
IM	1407	1492	1403	1491	1407	1492	1403	1491	11586
MG	1302	1302	1302	1302	1302	1302	1302	1302	10416
MS	1191	1580	1191	1577	1580	1577	1191	1577	11464
TR	1182	1669	1132	1564	1239	1657	1167	1622	11232
Total	7316	8543	7257	8427	7762	8528	7292	8485	63610

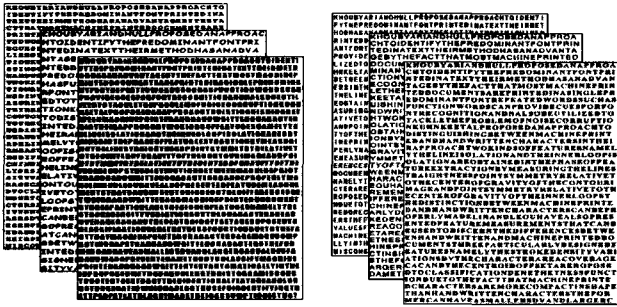


그림 4. 각 폰트에 대한 스캔된 문서들의 예; (a) CN, CS, IM 폰트영상, (b) MG, MS, TR 폰트영상

Fig. 4. Example of document images: from the left, (a) CN, CS, and IM font images, (b) MG, MS, and TR font images.

을 예시 하였다. [그림 5]는 Courier 폰트 'q' 와 Comic sans 폰트 'F'에 대해 각 다른 해상도에 대한 실패가 된 문자의 모양들을 보여준다. 그 모양들은 각 문자의 크기, 인쇄의 질 그리고 다른 해상도에 따라 다양하였다. 즉, 비록 같은 폰트의 같은 글자라 하더라도 공간적으로 다른 부분을 점령할 수 있다는 것을 보여준다.

[그림 6]은 랭크 수에 따라 실험에 사용된 베이스 집합을 보여준다. 훈련 단계에서 사용된 샘플 이미지들을 이용한 행렬분해 시, 각 샘플들이 모두 분해된 엔코딩들과 일대일 상응관계를 가지므로, 이를 이용하여 템플릿 엔코딩을 형성한다. [그림 6]에서 보여지는 것처럼 베이스 이미지의 수가 증가할수록 베이스 영상들과 이미지 엔코딩들은 더욱 희소해지고, 반대로 베이스 영상의 수가 감소할수록 하나의 베이스 이미지가 한 폰트의 지역적인 특징을 포괄하여야 하기 때문에

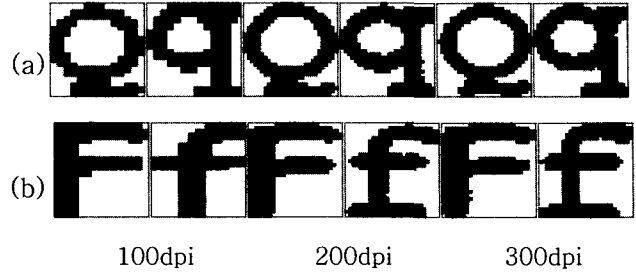


그림 5. 해상도에 따른 폰트의 모양 변이의 예. Fig. 5. Font shapes according to different resolution levels.

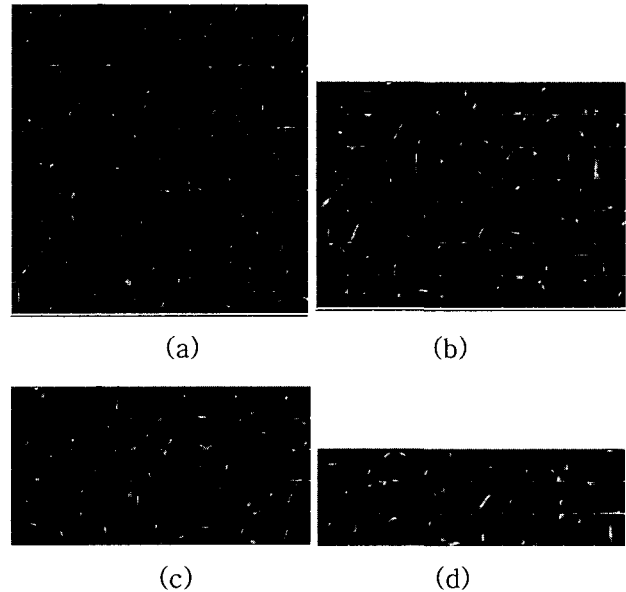


그림 6. 랭크 수에 따른 각 폰트의 국부적인 특징들 (a) 100 랭크 (b) 70 랭크 (c) 50 랭크 (d) 30 랭크 Fig. 6. Basis sets according to number of ranks: (a) 100 ranks, (b) 70 ranks, (c) 50 ranks, (d) 30 ranks.

베이스스 이미지와 각 폰트 이미지에 상응하는 엔코딩들의 희소성이 감소함을 볼 수 있다. 이처럼 랭크의 수에 따라 지역적인 특징을 학습한 베이스스의 모양이 달라지며, 본 논문에서는 부분적으로 랭크의 수에 따른 폰트 분류에 대하여 조사하였다.

[그림 7]은 폰트 영상을 폰트 영상들의 부분과 가중치(엔코딩)가 있는 베이스스의 조합으로 재구성한 예를 보여준다. [그림 7]에서 보여지는 것처럼, Courier New Capital 'A'의 원래 폰트 이미지는 [그림 7]의 (c)에서 보이는 것처럼 수식(6)을 이용하여 엔코딩으로써 가중치화된 베이스스 이미지들의 선형 결합을 통해 [그림 7]의 (b)처럼 근접된다. [그림 7]의 (c)와 같이 대부분의 가중치(엔코딩의 각 요소)가 0에 가까운 값이거나, 0이



9.957141	0.000000	10.036570	0.000000	0.000000	0.000000	9.600224	17.417137	11.494892	0.000000
0.000353	0.000000	6.716214	24.951897	0.000000	0.000000	4.185488	11.809018	0.000000	31.675209
0.000000	0.000000	0.000000	2.586174	0.000000	0.000000	0.000000	0.064374	0.000000	0.000000
0.000000	0.000000	0.000000	5.187489	8.492595	0.000000	0.000000	0.000000	9.955050	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.050783	12.461640
0.000000	0.000000	0.000000	1.518068	0.000000	0.000000	0.000000	13.071712	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.642389	0.000000	14.024639	17.361784	13.377803	10.866908
0.000000	0.000000	0.000000	0.084983	17.896767	0.000000	11.449212	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	14.749565	0.000000	0.000000	0.000000	0.104299	0.000000	22.755587
0.000000	12.454034	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

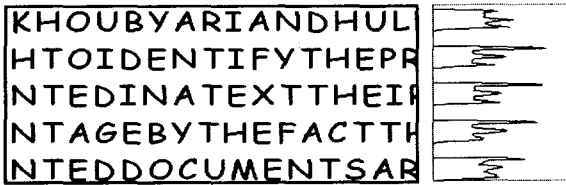
(c)

그림 7 베이스 집합과 한 엔코딩을 이용한 폰트 이미지의 재생성;

(a)원 영상, (b) 재구성된 이미지, (c) 베이스와 엔코딩의 선형 결합해상도에 따른 폰트의 모양 변이의 예  
 Fig. 7 Example of reconstruction using basis images and weighting coefficients; (a) original image, (b) reconstructed image, (c) basis images and corresponding coefficients for reconstructing Courier New Capital 'A' image.

기 때문에 한 폰트 이미지를 근접하여 재구성하기 위해 모든 베이스가 사용되지 않음을 알 수 있다. 이와 같이 각 베이스가 폰트의 각 부분을 학습하고 있고, 또한 그 부분들의 선형 결합이 하나의 폰트 이미지를 완

성할 수 있기 때문에 NMF 알고리즘이 폰트 이미지의 집합으로부터, 각 폰트의 국부적인 성질을 학습한다는 것을 알 수 있다. 따라서, 각각의 베이스 이미지를 서로 다른 가중치(엔코딩)를 사용하여 특정 폰트를 표현



(a)



(b)



(c)



(d)



(e)

그림 8. 전처리의 중간결과; (a) 수평 히스토그램을 이용한 텍스트 줄 분할, (b) 수직 히스토그램을 이용한 글자 단위 분할, (c) 글자단위 경계선, (d) 경계선 조절, (e) 크기 정규화 결과 이미지.

Fig. 8. Intermediate results of preprocessing; (a) text line separation, (b) character segmentation, (c) bounding boxes, (d) boundary adjustment, (e) size normalization.

하기 때문에, 이를 폰트를 분류하는 특징으로 사용하였다.

[그림 8]은 한 장의 문서 이미지로부터 각 글자들을 분리하여 입력 샘플을 생성하는 과정을 보여주고 있다. 인쇄된 문자들을 포함한 문서 영상들로부터 테스트 샘플을 얻기 위해서 다음의 문서 영상 처리 기술이 사용된다. 먼저, [그림 8]의 (a)처럼 문서 영상에서 문자열을 줄 단위로 분리하기 위해 수평 히스토그램을 이용하였다. 이는 대부분의 문서가 문자열과 문자열 사이에 일정한 여백이 삽입되어있음을 고찰한 결과이다. [그림 8]의 (b)에서는 (a)에서 생성된 한 줄의 글자 영상에서 글자 단위로 분리하기 위하여 수직 히스토그램을 사용하였다. 글자단위로 잘려진 영상들은 [그림 8]의 (c)와 같고, 각각의 글자 단위의 이미지의 경계를 조절하기 위해, 아래에서 위로, 위에서 아래로, 왼쪽에서 오른쪽으로 오른쪽에서 왼쪽으로 각각 경계선을 평행으로 이동시켰다. 이렇게 해서 생성된 문자 이미지들은 노이즈를 제거하

표 2. 분류 대상에 따른 폰트 분류.

Table 2. Classification rate (percent) of font types combined with each style.

	Regular		Bold		Italic		Bold Italic		Avg.
	Capital	Small	Capital	Small	Capital	Small	Capital	Small	
CN	99.1	98.5	98.67	98.7	100	99.7	100	99.6	99.28
CS	94.6	95.4	95.4	96.2	100	99.4	100	100	97.63
IM	94.4	95.2	94.46	95.6	98.0	100	100	98.8	97.06
MG	95.1	93	94.57	95.3	96.8	99.45	99.2	98.8	96.53
MS	90.2	90.4	96.7	96.8	98.8	98.4	100	99.2	96.31
TR	99.3	98.3	99.6	99.5	99.9	100	100	100	99.58
Avg.	95.45	95.13	96.57	97.02	98.92	99.49	99.87	99.4	97.73

기 위해 미리 정의된 크기 이상의 크기를 가지도록 크기 필터링을 거치고, 마지막으로 [그림 8]의 (e)와 같이 NMF의 입력으로 사용하기 위해 크기 정규화 과정으로 28×28 크기로 정규화 한다. 실제 실험에 사용된 이미지들은 글자는 흰색으로 글자가 아닌 부분(배경)은 검은색으로 표현된다. 실험 과정 동안 모든 글자들은 분할하기 쉽도록 글자사이의 간격이 분할하기 충분하다는 가정을 하였다. 또한 문서 전체 영상의 기울어짐 정도를 계산하지 않았다. 이 두 가정은 이전 연구의 결과를 이용한다면 충분히 해결 가능하다<sup>[9-12]</sup>.

각 엔코딩들이 폰트를 분류하기 위한 적절한 특징인지 여부를 검증하기 위하여, [그림 9]에서 보듯이 6개의 폰트들에 대한 엔코딩들의 평균을 도식화하였다. 각 엔코딩들은 각각의 폰트들을 구분하기에 충분히 다른 성질을 띄고 있다는 것을 알 수 있다. [그림 9]에서 X축은 원 행렬을 분해하기 위해 사용된 베이스스, Y축은 각 폰트별로 엔코딩들의 평균값을 표현한다. 이 그림에서 사용된 랭크의 수는 30 (r=30)이며, CN과 TR 폰트는 23번째 베이스스를 거의 사용하지 않고 있고, IM 폰트는 대부분의 베이스스를 비교적 높은 값으로 사용하고 있다는 것을 알 수 있다.

[표2]에서는 8개의 스타일을 가진 6개의 다른 폰트에 대한 분류률을 보여준다. 실험에 사용된 랭크의 수는 100이었다. 모든 테스트 데이터를 100랭크의 베이스스를 사용하여 사상하고, 이때 생성된 엔코딩과 훈련단계에서 생성된 템플릿을 비교한 폰트 분류률은 97.73%를 나타냈고, 그 중에 가장 낮은 분류률은 Microsoft Sansserif 폰트의 보통 글자체에서 나타났다. [표2]와 같



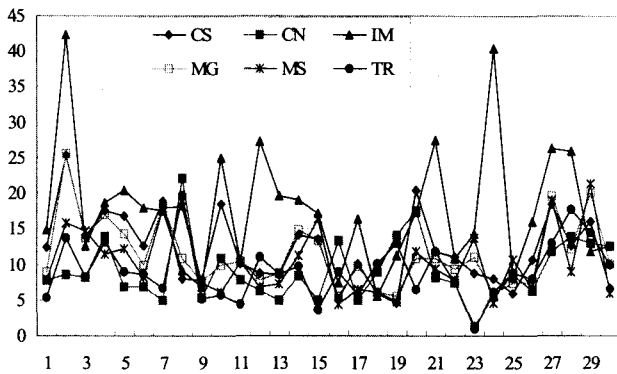


그림 9. 각 폰트에 상응하는 엔코딩들의 평균.  
Fig. 9. Six average encodings corresponding to each font.

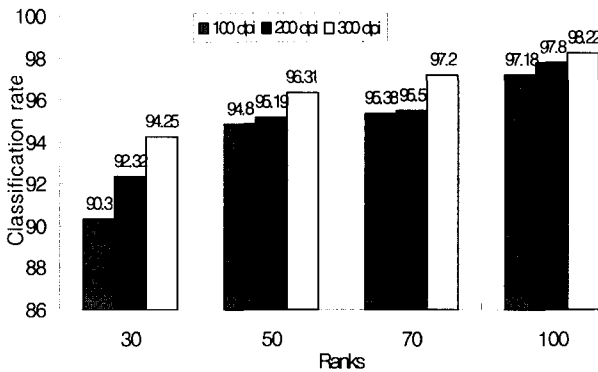


그림 10. 해상도의 차이에 따른 폰트 분류율.  
Fig. 10. Font classification rates (percent) according to different resolutions.

은 분류 정확도는 전역적인 인쇄상의 특징에 기반한 통계적 방법인 Zramdini와 Ingold의 방법<sup>[4]</sup>과 비교할 때, 특정 폰트인 TR폰트의 경우 Zramdini와 Ingold의 방법이 분류율 92.87%를 나타내는 반면 제안된 방법은 약 7%의 향상을 보인 99.58%의 정확도를 보였다. 하지만, CN 폰트의 경우는 Zramdini와 Ingold의 방법에서 99.30%의 정확도를 나타내는 반면, 제안된 방법은 99.28%로 0.02%의 차이를 보였다. 전역적인 특징을 사용하는 Zhu et al.의 방법에서는 TR폰트의 경우 평균 98.8%의 분류율을 보임으로써, 제안된 방법의 분류율 99.58%가 약 0.8% 정도 높은 결과를 보였다. 또한 CN 폰트의 경우, Zhu et al.의 방법은 100%의 정확도를 보이는 반면, 제안된 방법은 99.28%로 0.7%정도 분류율이 낮다. 그러나, Zhu et al.의 방법은 텍스트들로만 구성된 블록을 구성해야 하는 부담이 있고, 또한 블록 내에 포함된 여러 다른 폰트를 구별할 수 없는 단점이 있다.

표 3. 각 폰트별 오분류율.

Table 3. Typeface confusion matrix for each font type.

	CN	CS	IM	MG	MS	TR	MCR
CN	100	0.0	0.0	0.0	0.0	0.0	0.0
CS	0.0	98.5	0.0	0.4	1.1	0.0	1.5
IM	0.0	0.0	98.5	0.1	1.4	0.0	1.5
MG	0.0	0.7	0.6	97.2	1.5	0.0	2.8
MS	0.0	0.0	0.4	0.7	98.9	0.0	1.1
TR	0.0	0.0	0.0	0.0	0.0	100	0.0

[그림 10]에서는 30, 50, 70, 그리고 100 랭크가 각각 입력 폰트 행렬을 분해하기 위해 사용되었을 때, 폰트 분류율을 보여준다. [그림 10]에서처럼 전체적인 평균 분류율은 95.37%였다. 베이스이미지를 구하기 위한 학습단계는 off-line 처리과정을 거친다. 또한, 비 최적화 코드를 사용하여 하나의 테스트 문자를 분류하기 위해 소요된 시간은 30, 50, 70, 그리고 100 랭크 시에 각각 0.21초, 0.75초, 1.42초, 그리고 2.66초가 소요되었다. 이때, n의 크기는 784 (28×28)이고, 이는 다양한 폰트에서 각 문자들의 크기를 조사하여 실험적으로 설정하였다. 테스트 문자의 개수 m은 각 테스트마다 1이었다. r의 크기, 즉 베이스 이미지의 개수에 따라 시간 소요가 크다는 것을 각 랭크별 실험에 의해 알 수 있었다. 이는 하나의 테스트 문자에 대해 그 테스트 문자에 대한 새로운 엔코딩의 생성단계에서 많은 시간이 소요되었기 때문이다.

[표3]은 폰트 오류 행렬로써, 각 오류는 일반적으로 크기 정규화 과정이나, 문서의 스캔 과정에서 첨가된 잡음의 영향으로 분석된다. 표 내의 [i, j] 엔트리는 i 폰트의 입력이 j 폰트로 잘못 분류된 비율을 나타내고, 마지막 열은 전체적인 오분류율(misclassification rates, MCR)을 나타낸다. [표3]에서 보듯이 대부분의 오 분류는 같은 부류의 폰트들, 즉 장식이 없는 폰트체(Sanserif)에서 주로 발생하였고, 특히, MS, MG, 그리고 IM 폰트들의 소문자 'i' 와 'l' 와 MS, MG, 그리고 IM 폰트의 대문자 'I' 에서 주로 발생되었다. 반대로, 장식이 있는 폰트체(Serif)에서는 그런 종류의 오류는 범하지 않았다. 또한, 이러한 오류는 문서의 스캔 과정에서 첨가된 잡음의 영향도 있는데, [그림 8]의 (d)의 과정에서와 같이 경계선을 조절할 때 이러한 잡음들 때문에 올

바른 입력 샘플의 생성이 어려워진 경우가 발생하였다. 또한, 표 3에는 반영되지 않았지만, 각 폰트내의 굵은 폰트(bold type)와 일반 폰트(regular type)들 사이에서 많은 오류가 발생되었다. 따라서 이러한 오분류들은 각 폰트 이미지의 모양과 선의 굵기에 영향을 최대한 줄여주는 적절한 크기 정규화 알고리즘을 사용하고, 잡음을 적절히 제거하는 루틴을 첨가한다면 쉽게 해결될 수 있을 것이라 기대된다.

제안된 방법의 신뢰성을 위해 고문서(古文書)에 대한 실험과 한 OCR 시스템의 전처리기로써 폰트 분류에 대한 실험을 수행하였다. 먼저 고문서에 대한 실험으로, 1998년에 발행된 "IEEE Transaction on Consumer Electronics"의 한 문서에 대한 실험을 수행하였다. 실험에 사용된 실험영상은 저널 페이지에서 글자부분만을 수작업으로 추출한 후 [그림 11]의 (b) 이미지로 부터 자동으로 제안된 폰트 분류 방법을 사용하여 분류하였다. 680개의 연결 성분들로부터 전처리 단계를 거쳐 새로운 엔코딩을 생성하고, 이 엔코딩과 폰트의 템플릿들과 수식(7)을 이용하여 분류하였다. 분류 결과 Courier New 폰트로 73.42%분류되었다. 실제 테스트에 사용된 문서는 전적으로 Courier New 폰트는 아니었지만, 제일

표 4. 고문서 분류결과.

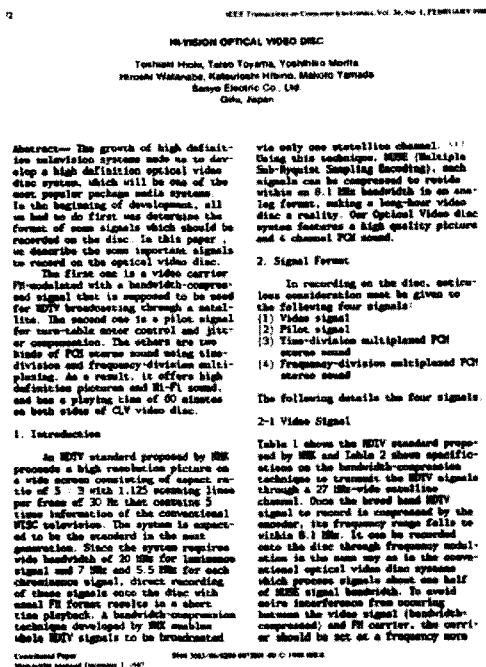
Table 4. Classification results on experiment for a paleography (%).

CN	CS	IM	MG	MS	TR
73.42	0	1.4	2.28	0	22.9

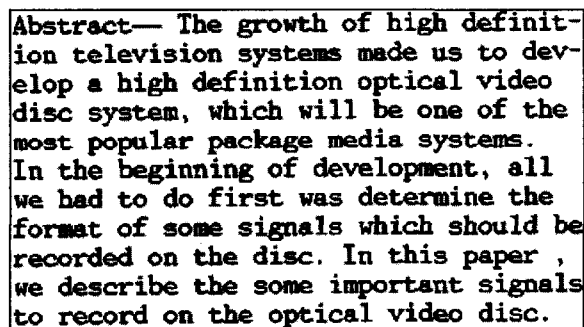
유사한 모양을 가진 Courier New 폰트로 가장 많이 분류되었다.

폰트 분류가 OCR 시스템의 전처리기로 사용될 때 그 성능이 부분적으로 향상됨을 입증하기 위해, 본 논문에서는 템플릿 매칭 (Template Matching) 방법을 이용한 OCR 시스템을 구현하였다. 템플릿의 구성은 베이스스 이미지의 학습단계에 사용된 4992개의 28×28 크기의 영상을 모두 사용하였으며, 템플릿 영상과 문서에서 추출한 문자영상의 유사도의 측정을 위해 식(8)과 같이 자카드 매치 (Jaccard measure)<sup>[17]</sup>를 사용하였다. 이는 템플릿의 영상 *i*와 테스트 영상 *j*간의 자카드 매치를 정의한 수식이다.

$$J_{ij} = \frac{b_{01} + b_{10}}{b_{00} + b_{01} + b_{10}} \quad (8)$$



(a)



(b)

그림 11. 고문서(古文書) 실험영상. (a) 전체 영상, (b) 부분영상

Fig. 11. Test sample image of paleography; (a) full image, (b) part of input images used to classify font.

여기서  $b_{01}$ 과  $b_{10}$ 은 한 영상에서는 객체의 일부분이고 다른 영상에서는 배경인 픽셀들의 수를 표현하고 있다. 즉, 두 영상에서 겹쳐지지 않은 부분의 픽셀 수이다.  $b_{00}$ 는 두 비교영상 모두에서 객체에 속한 픽셀들의 수이다. 이 자카드 매저는 두 이진 영상의 비유사도를 나타낸 값으로 그 결과값이 작을수록 잘 매칭되었다고 할 수 있다.

하나의 입력 문자 영상을 인식하기 위해 사용된 방법은 첫째, 4992개의 템플릿을 모두 사용한 경우의 실험 (실험 1)과 둘째, 먼저 폰트를 분류하고 분류된 폰트의 템플릿만을 대상으로 한 실험 (실험 2)이다. 실험 영상은 [그림 4]의 (f) 영상인 Times New Roman 폰트로 구성된 문서 영상을 사용하였고, 두 실험에 대한 OCR 결과는 다음 [표5]와 같다. [표5]에서 실험 2의 경우는 먼저 폰트가 분류된 다음에 템플릿 매칭을 통한 문자 인식과정이 수행되기 때문에 템플릿의 개수가 208개에 대한 비교 과정만을 거친다. 또한, [표5]의 결과로 off-line 문자 인식의 경우, 폰트의 분류를 선행함으로써 그 성능을 다소 향상시킬 수 있음을 보였다.

표 5. OCR 시스템의 인식률  
Table 5. Recognition rate for an OCR system (%).

	실험 1	실험 2
Recognition rate	98.6	99.03

### V. 결론

제안된 논문에서는 한 장의 영상을 부분의 결합으로 표현할 수 있는 NMF기술을 사용하여 다양한 폰트를 분류하는 새로운 방법을 제안하였다. 제안된 폰트 분류 방법은 한 폰트의 공간적으로 지역적인 특징을 학습하여 그 폰트의 특징으로 사용하는 방법이다. 실험은 문자 레벨로 실험되었고, 이러한 방식이 단어, 블록단위의 폰트 분류 시스템들보다 문자 인식 시스템의 전처리기에 보다 적합한 방법이다. 또한 만약 단어 단위의 실험을 사용하여 수행한다면 보다 좋은 분류 결과를 얻을 수 있을 것이라 기대된다. 본 논문의 초점은 NMF기술이 개별적 글꼴들을 표현하는 특징들을 학습할 수 있다는 것을 검증하는 것이다. 향후 과제로 NMF 공간에서의 랭크의 수와 거리척도에 대한 연구를 수행하여 좀더 일반화된 폰트 분류 방법을 만들기 위해 노력할 것이다.

끝으로, 제안된 방법이 기존의 OCR(optical character recognition)이나 문서 인덱싱 및 검색 시스템들의 전처리로 사용되면, 그 시스템들의 성능을 향상시킬 것으로 기대된다.

### 참고 문헌

- [1] G. Nagy, "Twenty Years of Document Image Analysis in PAMI," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 38-62, 2000.
- [2] S. Khoubyari, J. J. Hull, "Font and function word identification in document recognition," Computer Vision and Image Understanding, vol. 63, no. 1, pp. 66-74, 1996.
- [3] H. Shi, T. Pavlidis, "Font Recognition and Contextual Processing for More Accurate Text Recognition," Proceedings of Document Analysis and Recognition '97, pp. 39-44, 1997.
- [4] A. Zramdini, R. Ingold, "Optical Font Recognition Using Typographical Features," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 877-882, 1998.
- [5] Y. Zhu, T. Tan, Y. Wang, "Font Recognition Based on Global Texture Analysis," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1192-1200, 2001.
- [6] D. D. Lee, H. S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," Nature 401, pp. 788-791, 1999.
- [7] H. S. Seung, "Derivation of the objective function (Eq.2)," <http://journalclub.mit.edu>.
- [8] D. D. Lee, H. S. Seung, "Algorithms for non-negative matrix factorization," In Advances in Neural Information Processing Systems, 13, pp. 556-562, 2001.
- [9] Y. Lu, "Machine printed character segmentation - an overview," Pattern Recognition, vol. 28, no. 1, pp. 67-80, 1995.

- [10] J. H. Bae, K. Jung, J. W. Kim, H. J. Kim, "Segmentation of Touching Characters Using an MLP," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 701-709, 1998.
- [11] K. Jung, "Neural network-based Text Location in Color Images," *Pattern Recognition Letters*, vol. 22, no. 14, pp. 1503-1515, 2001.
- [12] B. Yu, A. K. Jain, "A Robust And Fast Skew Detection Algorithm for Generic Documents," *Pattern Recognition*, vol. 29, no. 10, pp. 1599-1629, 1996.
- [13] C. W. Lee, H. Kang, K. Jung, H. J. Kim, "Font Classification Using NMF," *Lecture Notes in Computer Science 2756*, pp. 470-477, 2003.
- [14] I. Biederman, "Recognition-by-components: A theory of human understanding," *Psychological Review*, vol. 94, no. 2, pp. 115-147, 1987.
- [15] L. Shams, *Development of Visual Shape Primitives*, PhD thesis, University of Southern California, 1999.
- [16] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," In *Proc. of 6th European Conference of Computer Vision*, 2000.
- [17] A. K. Jain and D. Zongker, "Representation and Recognition of Handwritten Digits Using Deformable Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1386-1391, 1997.

---

 저 자 소 개
 

---



이 창 우(정회원)

1998년 경북대학교  
컴퓨터공학과 공학석사  
1999년~2001년: 포항1대학 교수  
2004년 경북대학교  
컴퓨터공학과 공학박사

<주관심분야 : 비전인식, 패턴 인식, 비디오 인덱싱, Video Retrieval, 문자 검출>



강 현(정회원)

1999년 경북대학교  
컴퓨터공학과 공학석사  
2000년~2001년 (주)네비웍스  
책임연구원  
2004년 경북대학교  
컴퓨터공학과 공학박사

<주관심분야 : 게임, HCI, Augmented Reality, 이미지 처리, 제스처 인식, 문자 인식>



정 기 철(정회원)-교신저자

1996년 경북대학교  
컴퓨터공학과 공학석사  
2000년: 경북대학교  
컴퓨터공학과 공학박사  
2001년~2002년 (미) 미시간대  
Anil K. Jain 교수 PRIP  
연구실 박사 후 연구원,

2003년~현재 숭실대학교 미디어학부 교수  
<주관심분야 : 문자 인식, 이미지 처리, 패턴 인식, 비디오 인덱싱, Augmented Reality, Mobile Vision System>

김 항 준(정회원)

제40권 CI편 제6호 참조