논문 2004-41SP-6-20

복잡한 영상 내의 문자영역 추출을 위한 텍스춰와 연결성분 방법의 결합

(Hybrid Approach of Texture and Connected Component Methods for Text Extraction in Complex Images)

정 기 철*

(Keechul Jung)

요 약

본 논문은 복잡한 컬러 영상에서의 문자 추출을 위한 텍스춰와 연결성분 방법의 결합된 방법을 제안한다. 자동 학습 방법으로 구축된 다층 신경망(multilayer perceptron)은 부트스트랩 학습 방법을 사용함으로써 별도의 특징값 추출 단계 없이 다양한 환경의 입력 영상에 대한 검출률(recall rate)을 향상시키며, 검출률을 향상함으로써 발생되는 정확도(precision rate) 저하 문제는, NMF(Non-negative matrix factorization)를 이용한 연결 성분 방법을 사용함으로써 극복한다. 문자의 존재 비율이 낮은 입력영상에 대하여 CAMShift 알고리즘을 이용한 영역 마킹 방법을 사용함으로써, 두 방법을 결합함으로써 야기되는 속도 저하 문제의 해결을 시도하였다. 이와 같이 텍스춰와 연결성분 방법을 결합함으로써 강건하고 효율적인 시스템을 구성할 수 있었다.

Abstract

We present a hybrid approach of texture-based method and connected component (CC)-based method for text extraction in complex images. Two primary methods, which are mainly utilized in this area, are sequentially merged for compensating for their weak points. An automatically constructed MLP-based texture classifier can increase recall rates for complex images with small amount of user intervention and without explicit feature extraction. CC-based filtering based on the shape information using NMF enhances the precision rate without affecting overall performance. As a result, a combination of texture and CC-based methods leads to not only robust but also efficient text extraction. We also enhance the processing speed by adopting appropriate region marking methods for each input image category.

Keywords : Text Extraction, MLP, Texture, NMF(Non-negative Matrix Factorization), Connected Component(CC), CAMShift

I. 서 론

전통적으로 영상 데이터는 많은 시간과 노동력을 요구하는 수작업에 의해 인덱싱되거나, 상대적으로 저 차원적인 색상, 모양, 질감 등의 정보를 이용하여 자동으로 인덱싱되고있다. 최근에는 영상에 내재된 문자를 추출하고 인식함으로써 인덱싱에 유용한 고급 정보를 얻을 수 있고, 또한 다른 내용 기반 정보(content-based information)에 비해 상대적으로 추출이 용이하다는 장점으로 인하여, 영상 내의 문자 추출에 관한 연구가 멀티미디어 시스템, 전자 도서관, 비디오 인덱싱, 문서 구조 분석, 우편 영상 내의 주소 영역 추출, 자동차 번호판 추출 등 다양한 관련 분야에서 진행되고 있다. 그러나 양질의 문서의 자동 문자 인식(OCR) 과는 달리 복잡한 배경, 다양한 글자 모양으로 인해 영상 내의 문자 추출은 상당히 어려운 문제로 인식되고 있다[1-16].

기존의 문자 추출 방법은 크게 연결 성분 방법(connected component-based method)과 텍스춰 (texture-based) 방법으로 나눌 수 있다. 연결 성분 방법은 색

* 정회원, 숭실대학교 정보과학대학 미디어학부
(School of Media, College of Information Science, Soongsil University)
※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음
접수일자: 2003년10월9일, 수정완료일: 2004년11월4일

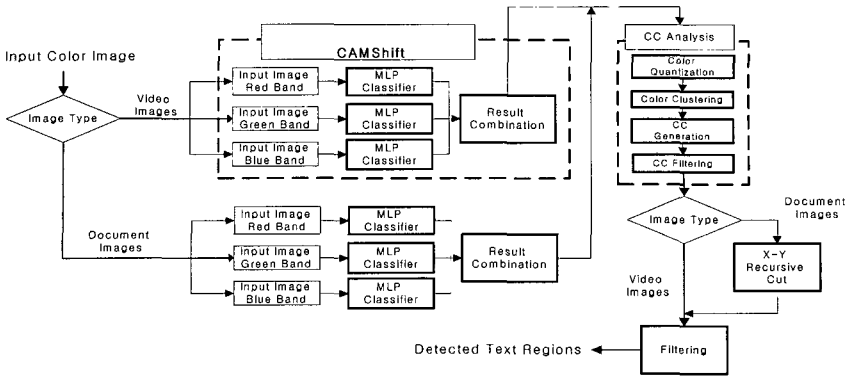


그림 1. 전체적인 시스템 구성도
Fig. 1. Overview of proposed approach.

상, 질감 등의 정보를 이용하여 입력 영상을 분할한 후, 각 연결 성분(connected component:CC)들을 특정 조건들을 이용하여 문자 영역과 비-문자 영역으로 나누는 방법이다[1, 5, 6]. Lienhart와 Studer[1]는 비슷한 색상, 크기의 연결 성분을 문자로 간주하여 추출하고 모션 정보를 이용하여 추출 결과를 향상시켰다. Jain과 Yu[5]는 입력 프레임을 각기 다른 색상으로 분할한 후, 각 분할된 조각들을 크기나 배열 모양 등의 조건들을 이용해서 걸러냄으로써 문자 추출을 수행한다. Kim[6] 등은 한글과 한자 등의 다-조각 문자 (multi-segment character)를 포함하는 영상에서 비-문자 성분들을 걸러내기 위해 클러스터 기반 템플릿을 사용하였다.

연결 성분 방법과 달리 텍스춰 방법은 문자 영역의 텍스춰 성질을 이용하기 위해 gabor filter, wavelet, spatial variance 등의 텍스춰 분석기를 사용하는 방법이다[3, 4, 7, 8, 11-15]. Li[3] 등은 wavelet을 이용한 특징값 추출 후 신경망을 이용하여 문자 영역을 추출하였다. Zhong[4] 등은 DCT 압축된 도메인에서 직접 텍스춰 특징을 이용한 방법을 사용하였다. Zhong[7] 등은 그레이 영상에서의 국소 영역 변화 (local spatial variation)가 높은 차이를 보이는 영역을 문자 영역으로 간주하였다. Jain과 Karu[18]는 서류 영상 내의 문자, 그래픽, 반 음영 영역 (halftone region)을 구분하기 위해 학습 기반의 텍스춰 분석 기법을 사용하였다. Jung[30]은 자동화된 학습 기반 신경망을 이용하여 영상 내의 문자의 텍스춰 성질을 분석하여 문자를 추출하였다.

문자 추출을 위한 일반적인 방법인 연결 성분 방법은 구현이 쉬운 반면, 문자 크기와 문자 간의 거리 등의 휴리스틱한 정보에 많은 영향을 받으며, 정밀한 영상 분할 알고리즘을 필요로 하기 때문에, 비디오 영상과 같은 잡음이 많은 저해상도 영상 또는 다양한 크기의 영상들에는 적합하지 않다. 또한 문자 추출에 상당히 효과적이기는 하지만 텍스춰 방법 또한, 텍스춰 분석기 생성의 어려움, 텍스춰 분석 단계에서의 많은 계산량, 텍스춰 분석 후의 영역 마킹(marking) 과정에서 전체적인 시스템의 성능 저하와 같은 몇가지 단점들이 있다.

본 논문에서는 텍스춰 방법과 연결 성분 방법을 결합한 문자 추출 방법을 제안한다. 제안한 방법은 다음과 같은 각 방법의 장점을 이용한다. 문자 영역과 비-문자 영역을 분류하는 텍스춰 분석기를 생성하기 위해 MLP(multi-layer perceptron)를 사용함으로써 다양한 환경에 적응성을 지니는 분석기를 자동으로 생성한다. 부트스트랩 방법[17]을 이용하여 MLP의 검출률(recall rate)을 향상시키고, 이에따른 불가피한 오추출(false alarm)의 증가는, 기존의 기하학적인 연결 성분의 배치 관계를 이용한 필터링 방법보다 정확한 필터링을 수행함으로써 정확도(precision rate)를 적정선으로 유지한다. MLP의 계산 결과는 미디언필터링, 색상양자화 등의 단계를 통한 후, 연결성분의 모양, 위치 정보 등을 이용하여 필터링하게 된다. 필터링을 위해서 여러 휴리스틱 정보에 대한 최대 우도값 분류기 (maximum likelihood classifier)를 사용하며, NMF (Non-negative Matrix Factorization) 기반의 연결 성분 필터링 방법을 이용하여 기존의 기하학적인 연결 성분의 배치관계를 이용한 방법보다 정확한 필터링을 수행한다.

이와 같은 텍스춰 방법과 연결 성분 방법을 결합하여 사용함으로써 검출률과 정확도를 향상할 수 있지만, 수행시간이 증가되는 단점이 있다. 이러한 수행시간의 대부분은 텍스춰 분석 단계에서 사용되는데, 본 논문에서는 텍스춰 분석 단계에서의 많은 계산량으로 인한 속도 저하를 CAMShift 알고리즘을 사용하여 불필요한 부분에 대한 처리를 하지 않음으로써 해결하고, 문서 영상

에 대해서는 X-Y 재귀 알고리즘을 이용하여 문자영역을 마킹한다. 문서 영상과 같이 문자 영역의 존재 비율이 많은 영상에 대해서는 CAMShift 알고리즘의 효과가 그리 크지않지만, 비-문자 부분에 비해서 상대적으로 적은 문자 영역을 포함하고 있는 비디오 영상에 대해서는 상당히 수행 속도가 향상된다. 전체적인 시스템의 구성은 그림 1과 같다.

본 논문의 구성은 다음과 같다. 제 II, III장에서는 텍스춰 방법, MLP, 연결 성분 방법을 설명하고, X-Y 재귀 분할 알고리즘과 CAMShift 알고리즘을 이용한 필터링은 IV장에서, 제 V장에서는 제안된 방법을 이용한 실험 결과와 분석을 보이고, 제 VI장에서는 결론과 향후 연구 분야를 기술한다.

II. 텍스춰 분석 MLP

문자 영역과 비-문자 영역을 구분하는 텍스춰 분석기를 구성하는 어려움을 극복하기 위해 학습을 통한 텍스춰 분석기 생성에 관한 연구가 진행되고 있는데[3, 8, 12, 18], 본 논문에서는 MLP를 사용하여 다양한 크기나 모양의 문자와 배경에 적응할 수 있는 텍스춰 분석기를 구성한다[30, 31].

MLP는 2개의 은닉층, 1개의 출력 노드로 구성되며 인접층의 노드들은 모두 연결되어있고, 입력 영상에서 M×M 크기의 입력창 내의 화소들의 색상값을 사용한다. MLP를 이용하여 입력 영상을 처리 함으로써 생성된 MLP의 출력 영상(TPI: Text Probability Image)의 각 화소는 대응하는 입력 영상의 각 화소를 문자와 비-문자 클래스로 구분한다. 이 영상은 [0..1] 사이의 값을 지니며, 각 값은 입력 영상 내의 해당하는 화소의 문자 여부를 나타낸다. 본 연구에서는 컬러 영상을 Red, Green, Blue 세가지의 색상 밴드로 분할한 후, 동일한 모양의 MLP를 이용하여 문자 여부를 계산한 후, 이들을 중재 신경망(arbitration network)을 통해 최종 결과값을 생성한다[13, 31]. 신경망의 학습 단계에서 비-문자 클래스 학습을 위해 부트스트랩 (bootstrap) 방법을 사용한다[17]. 이는 초기에 정해진 비-문자 클래스 데이터를 이용해서 신경망을 학습 시킨 후에, 부분적으로 (partially) 학습된 신경망을 테스트 영상에 테스트하여 오인식 된 데이터로 신경망을 재 학습시키는 방법이다. 이를 통하여 보다 정교한 학습이 가능하게된다.

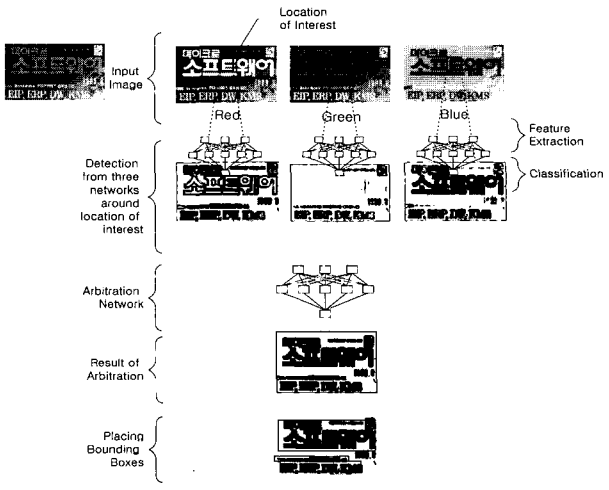
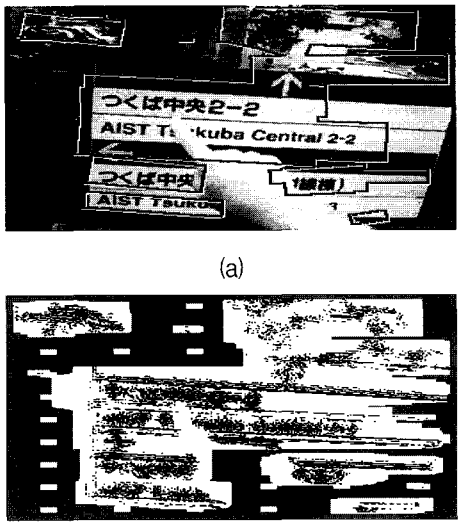


그림 2. 텍스춰 분석기 MLP네트워크의 구조
Fig. 2. Architecture of a discrimination network using multiple text detection networks.

III. 연결 성분 기반 문자 필터링

MLP가 문자와 비-문자의 경계를 명확하게 학습하기 위해 부트스트랩 방법을 사용하더라도, 학습을 통한 신경망은 여전히 많은 비-문자영역(false alarm)을 가지게 된다.

그림 3은 MLP를 이용한 문자 영역 추출 결과이다. 그림 3의 (b)에서 폐곡선으로 마킹된 영역 내의 검은색 화소들은 MLP에 의해 문자 영역으로 표시된 부분이며, 흰부분은 비-문자 부분, 마킹된 영역 밖의 회색 부분은



(a)

(b)

그림 3. MLP를 이용한 문자 추출 예: (a) 마킹된 문자 영역, (b) 신경망의 결과 영상
Fig. 3. Text detection examples using MLP: (a) image with marked text regions and (b) its corresponding MLP output.

신경망을 적용할 필요가 없는 부분이다. 그림 3처럼 색상 변화가 심한 영역에서 MLP는 많은 오추출(false alarm)을 보이는데, 기존에는 간단히 문자 영역의 크기나 가로 세로 비율을 이용하여 필터링을 하였으나, 이를 완전히 제거하기는 쉽지 않다. 이러한 오추출은 MLP가 영상의 제한된 크기의 서브 영역만을 고려하는 국부적 특성에 기인한다.

또한 MLP의 결과는 문자영역의 위치만을 표시하여 준다. 즉, MLP는 찾은 문자열을 배경과 분리(추출, extraction)하여 주지는 않으며, 이후의 문자인식 단계에서 배경과 분리된 추출된 문자를 요구할 때, 별도의 작업이 필요하게 된다. 지금까지 대부분의 텍스춰 기반 문자 추출 방법들이 이 문제를 거의 언급하지 않고있으며, 연결 성분 방법과의 결합된 방법을 사용함으로써 비로써 문자의 추출이 이루어진다고 볼 수 있다.

MLP의 결과 영상은 5□5의 미디언 필터링을 거친 후, 각 Red, Green, Blue 색상별 3비트만을 남겨둠으로서 512 레벨로 양자화한다. 영상 내의 각 색상들은 single-link 클러스터링 알고리즘[35]을 통해서 자신의 대표 색상으로 클러스터링된다. 각 클러스터 간의 거리를 효과적으로 계산하기 위해, 두 색상을 병합할 때 많은 분포를 가진 색상을 대표 색상으로한다. 실험을 통한 결과, 보통 6-9개의 대표 색상으로 클러스터링된다. 영상 내의 각 픽셀은 자신의 대표 색상으로 표현한 후, 각각의 색상 공간에서 연결 성분을 구하고 3□3 블록마스크를 이용한 클로징 모폴로지 연산을 수행한 후, 연결 성분들의 특성값(크기, 면적, 가로 세로 길이 등)을 추출한다. 이러한 전처리 과정을 거친 후 다음의 3단계 필터링을 수행한다.

Stage 1. 각 연결 성분의 크기, 채워진 면적 비(fill factor), 가로세로 비율: 각 클래스(문자 클래스, 비-문자 클래스)의 가우시안 분포(Gaussian densities)를 이용한 최대 우도 분류기(maximum likelihood classifier)을 사용한다. 각 연결 성분의 크기와 면적, 가로 대 세로 길이비, 채워진 면적비 등을 Gaussian 분포로 모델링하기위해서, 학습 데이터로부터 이들의 평균 벡터(mean vector)와 공분산 행렬(covariance matrix)을 추정한다. 각 특성값 별 우도값(likelihood)은 획득된 연결 성분 데이터를 이용해서 아래의 식과 같이 나타낸다. 식에서 첨자 t와 nt는 각각 문자와 비-문자를 나타낸다. 신경망을 이용하여 필터링된 컴포넌트들 중 문자와 비-문자 컴포넌트를 수작업으로 구분한 후 이를 학습 데이터로 사용한다. 문자 클래스와 비-문자 클래스 각각 독립적으로 수행하여 최대우도값을 가지는 클래스를 선택한다.

$$p(size\_x, size\_y, fill\_factor, x\_y\_ratio, area \mid text) \approx N(\mu_t, \Sigma_t) = \frac{1}{\sqrt{2\pi}^5 \sqrt{|\Sigma_t|}} e^{-\frac{1}{2}(x-\mu_t)\Sigma_t^{-1}(x-\mu_t)}$$

$$p(size\_x, size\_y, fill\_factor, x\_y\_ratio, area \mid non\text{-}text) \approx N(\mu_{nt}, \Sigma_{nt}) = \frac{1}{\sqrt{2\pi}^5 \sqrt{|\Sigma_{nt}|}} e^{-\frac{1}{2}(x-\mu_{nt})\Sigma_{nt}^{-1}(x-\mu_{nt})}$$

Stage 2. 연결 성분의 배치: 같은 문자열에는 2개 이상의 연결 성분이 위치해야한다. 이웃한 2개의 연결 성분들을 통합하는 MBR(minimum bounded rectangle)의 가로 대 세로의 비율이 특정크기 이상이어야한다.

Stage 3. 모양 정보: Stage 1과 2만으로는 연결 성분을 명확히 구분하기는 쉽지않다. 최근의 연구 결과[29]에서 Stage 1과 2 단계와는 별도로 클러스터기반 템플릿(그림 4)을 이용한 필터링 기법을 사용하였다. 이 방법은 기하, 위치 정보를 통해서 K-means 등의 클러스터링 알고리즘을 이용하여 구성된 템플릿을 이용하여, 필터링된 연결 성분들을 문자의 모양 정보를 이용해서 세밀하게 필터링하는 단계이다[29]. 그러나 이 방법은 다양한 글자체와 글자 크기에 따른 변형을 흡수하기에는 미흡하다. 본 논문에서는 다양한 글자체의 변형을 흡수하기 위해서 NMF 기반의 필터링 방법을 제안한다.

Lee and Seung[32-34]에 의해 제안된 NMF 알고리즘은 주어진 매트릭스를 베이시스 매트릭스와 인코딩 매트릭스로 분할 한다. 이때 베이시스, 인코딩 매트릭스는 비-음수(non-negative)이다. 이와 같이 NMF는 V를 분할하여 n × r 베이시스 W와 r × m 인코딩 매트릭스 H로 다음의 식 (1)과 같이 분할한다.

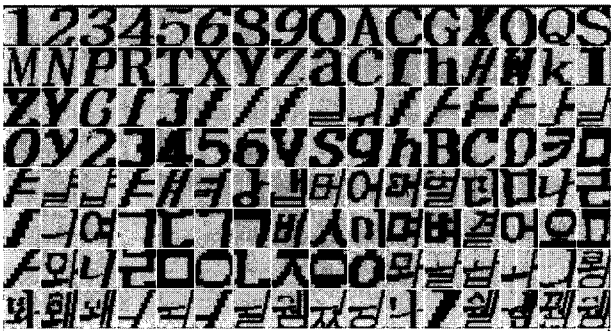


그림 4. 클러스터 기반 템플릿
Fig. 4. Cluster-based template examples.

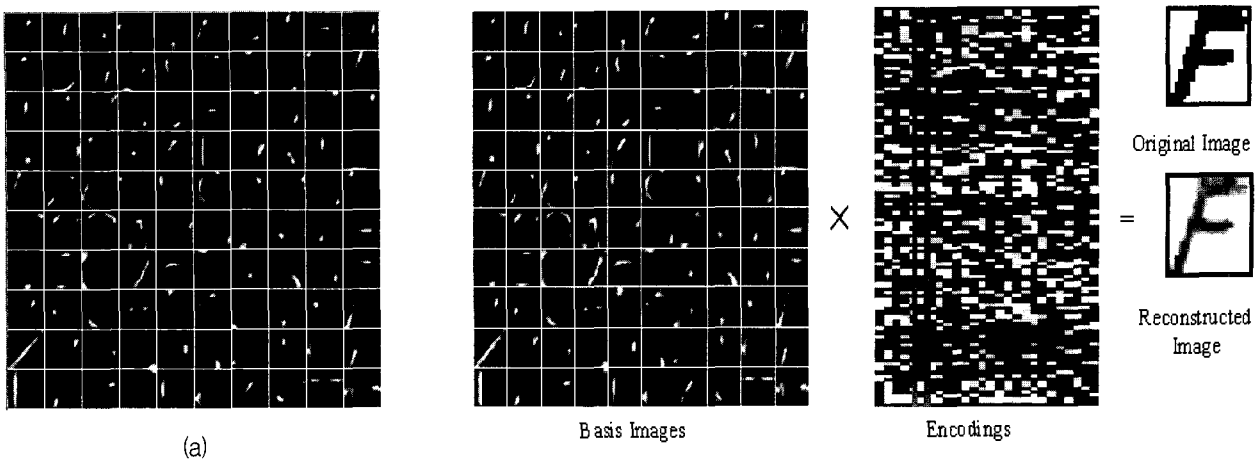


그림 5. 베이시스 이미지와 인코딩 예: (a) 베이시스 이미지, (b) 인코딩 예
Fig. 5. Basis images and encodings: (a) basis images, (b) encodings.

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

where $(n+m)r < nm$. (1)

이러한 NMF 알고리즘은 분할된 두 매트릭스의 직관적인 정보표현 능력으로 인해, 영상처리 분야에서 최근에 많이 사용되는 방법이다. 본 연구에서는 일반화된 문자 모양을 표현하기 위해서 입력된 연결 성분들을 랜덤하게 초기화된 W와 H를 이용하여 매트릭스 분할을 수행하는데, 수식 2와 같은 곱셈 수정 규칙(multiplica -tive update rules)에 의해서 W와 H를 반복적으로 수정한다.

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

$$W_{ia} \leftarrow W_{ia} \frac{W_{ia}}{\sum_j W_{ja}}. \quad (2)$$

그림 5에 이를 통한 베이시스 영상들을 예를 들어 보였다. 각각 28 × 28 픽셀 크기로써 [0..255]사이의 값들로 정규화되었다. 학습을 위해 사용된 문자 영역과 비-문자 영역들에 대한 인코딩을 구한 후, 이를 문자영역과 비-문자 영역 각각을 하나의 은닉층을 가진 MLP를 이용하여 학습한다.

베이시스 W가 특정 데이터 클래스를 표현하기위한 특징들을 학습한 후, 수식 (3)과 같이 베이시스 행렬 W는 그대로 사용하여 새로운 테스트 벡터 v를 NMF 공간으로 사상 시킨다. 이때, 임의로 초기화 되어진 H 행렬만을 갱신함으로써 테스트 벡터 v에 상응하는 인코딩 h를 구할 수 있다. 이러한 방식으로 입력 영상에 상응하는 새로운 인코딩 h를 구할 수 있고, 결과적으로 베이시스 W는 갱신없이 사용되면서 새로운 인코딩 h를 계산할 수 있다. 이렇게 구해진 인코딩은 학습 단계에서 구해진 문자와 비-문자들에 대한 인코딩을 학습한 신경망에 의해 문자와 비-문자를 구분한다.

$$v_{n\times 1} \approx (W_{n\times r}) h_{r\times 1} = \sum_{a=1}^{r} \sum_{i=1}^{n} W_{ia} h_{a1}. \quad (3)$$

Ⅳ. 영역 마킹

입력 영상의 종류에 따라서 두 가지의 문자영역 마킹 방법을 사용한다. 지금까지 많은 연구가 문자 검출(text detection)에 대해서 이루어지고있지만, 문자 검출 후의 추출(extraction)에 관한 연구는 그리 많이 행해지지 않았다. 본 논문에서는 컬러 문서 영상에 대해서는 기울기 보정이 사전에 되어 있다는 가정하에 X-Y 재귀 분할 알고리즘을 이용하였고, 문자 영역의 비율이 상대적으로 적은 비디오 영상에 대해서는 텍스춰 분석 단계에서 CAMShift 알고리즘을 사용하였다. 이렇게 각 입력 영상에 맞는 영역 마킹 알고리즘을 사용함으로써 수행시간과 성능면에서 효율성을 기할 수 있었다. 또한 최종적으로, 위의 두 방법으로 마킹된 영역의 크기와 가로 대 세로의 비를 이용하여 최종 필터링을 수행한다.

4.1. X-Y 재귀 분할 알고리즘

문자 영역이 상대적으로 많이 존재하는 문서 영상에

대해서, X-Y 재귀 분할 알고리즘을 이용하여 문자 영역을 마킹한다. 이 알고리즘은 탑-다운(top-down) 방법으로 재귀적으로 수행되는 알고리즘으로써, 이진 영상(문자 영역을 검은색, 비문자 부분은 흰색)을 입력받아, 가로, 세로 방향으로 프로젝션된 프로파일의 굴곡을 찾아서 재귀적으로 분할한다. 입력 문서는 사전에 기울기 보정이 되어있다고 가정한다. 그림 6은 재귀 분할 알고리즘이며, Split_by_X_Projection(region)은 그림 6과 유사하다. 그림 7과 8은 문서 영상의 X-Y 재귀 분할 알고리즘 수행 결과이다.]

4.2. CAMShift 알고리즘

문서 영상과는 달리 비디오 영상에는 문자의 존재 비율이 상당히 낮은 편이다. 본 논문에서는 텍스춰 분석 단계에서 MLP의 결과 영상(TPI) 상에 위치한 다수의 검색창에서 CAMShift 알고리즘을 반복 수행함으로써 영상 내의 문자를 검출한다[31]. 시작 단계에서 영상 내의 각 검색창이 문자 영역을 포함하고 있는지를 결정하며 (text detection), 연속된 일련의 단계에서 2차원 모멘트 (moment) 계산을 통해서 문자 영역의 크기와 위치를 구하고 인접한 화소를 병합하면서 문자 영역을 찾게 된다(text localization). 매 번의 반복 수행과정에서 검색창의 크기를 문자 영역의 크기에 비례해서 수정한다. 이 방법은 상대적으로 문자 영역이 비-문자 영역에 비해 적은 영상에서, 전체 입력 영상의 탐색 없이 필요한 부분만을 스캔 함으로써 전체적인 수행속도를 개선할 수 있다(그림 9). 본 방법은 CAMShift 알고리즘 내에 MLP를 이용한 필터링 연산이 포함된 형태이다. 문자 영역의 위치와 크기를 구하기 위해 연산이 간단하고 잡음에 효과적인 모멘트를 사용하였다. 이차원 p+q 차 모멘트는 다음과 같이 기술할 수 있다.

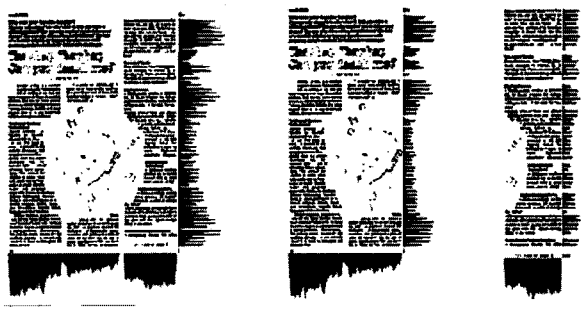
$$M_{pq}(i) = \sum_{x}\sum_{y} x^{p} y^{q} TPI\ (x, y) \tag{4}$$

문자 영역의 중심 좌표 (sample mean location)는 평면 커널을 사용할 때

$$mean_{X}(i)_{t} = \frac{M_{10}}{M_{00}}, \quad mean_{Y}(i)_{t} = \frac{M_{01}}{M_{00}}. \tag{5}$$

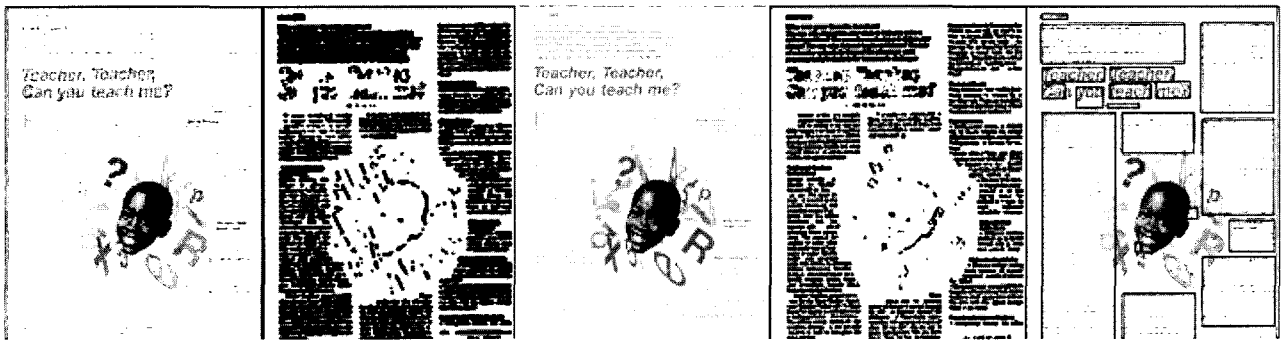
```
Function Split_by_Y_Projection(region)
{
    Project_on_Y_Axis(region);
    Find_Valleys_in_Projection;
    If (valleys satisfy threshold)
    {
        Split_Region_At_Valleys;
        For(each sub-region from splitting)
                Split_by_X_Projection(region);
    }
}
```

그림 6. X-Y 재귀 분할 알고리즘
Fig. 6. X-Y recursive cut algorithm.



(a) (b) (c)

그림 7. X-Y 프로젝션의 예: (a)x, y 프로젝션, (b,c) 각 분할된 영역별 프로젝션
Fig. 7. Projections for identifying text regions (a)x and y projections, (b,c)projections for two sub-regions.



(a) (b) (c) (d) (e)

그림 8. X-Y 재귀 분할 알고리즘: (a)입력, (b)MLP 출력, (c)색상 양자화, (d)CC 필터링, (e)최종 문자 추출 결과
Fig. 8. X-Y recursive cut: (a)input image, (b) MLP's output, (c)color quantized image, (d) CC analysis, and (e) text regions.

- Set up the initial locations ($mean_X(i)_0$, $mean_Y(i)_0$) and sizes ($\lambda_x(i)_0$, $\lambda_y(i)_0$) of search windows Ws on the image depending on the applications.
- Do
  For each window W(i)
  {
  - Generate the TPI within W(i) using MLP.
    for all pixel(x, y) in an image, where $\| x - mean_X(i)_t \| \leq \lambda_x(i)_t \; \| y - mean_Y(i)_t \| \leq \lambda_y(i)_t$
    if pixel(x,y) has been convolved by MLP in former iterations, re-use TPI(x,y),
    else perform convolution using MLP at pixel(x,y).
  - Based on the mean shift vector, derive the new location and size of a text region.
  - Move the W(i) to the new location and change its size.

  }
  Merge overlapping nodes.
  Increment the iteration number t.

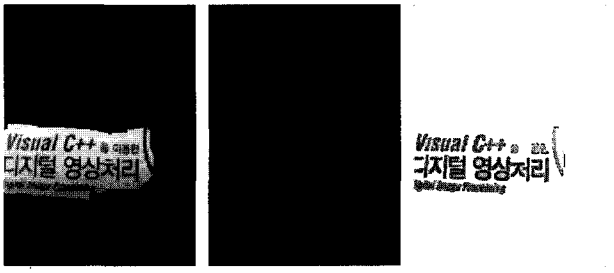
While ($\| mean_X(i)_t - mean_X(i)_{t+1} \| > \varepsilon_x$ or $\| mean_Y(i)_t - mean_Y(i)_{t+1} \| > \varepsilon_y$)

그림 9. CAMShift 알고리즘
Fig. 9. CAMShift algorithm.

그림 10. 검색창의 변화에 따라 문자 검출(좌상단부터)
Fig. 10. Transition of convolved regions using CAMShift: from left top to right bottom.

로 설정할 수 있고, 구해진 중심 좌표 값에 따라서, 입력 영상에서의 문자 영역을 CAMShift를 이용하여 탐색하는데, mean shift 값이 정해진 임계값 이하일 때까지 반복 수행한다.

문자 영역의 폭 (width)과 높이 (height)는 문자 영역에 대응하는 사각형 영역의 가로, 세로 길이에 해당하는 것으로, 2차원 모멘트로 구성된 행렬의 고유값(eigen value)를 구함으로써 다음과 같이 표현할 수 있다[23].



(a) (b) (c)

그림 11. CC 기반 필터링: (a) 텍스춰 분석, (b)양자화, (c) CC 기반 필터링
Fig. 11. CC-based filtering: (a) texture analysis, (b) quantization, and (c) CC-based filtering.

$$width\ (i)_t = \sqrt{2(a+c) + 2\sqrt{b^2 + (a-c)^2}}$$

$$height\ (i)_t = \sqrt{2(a+c) - 2\sqrt{b^2 + (a-c)^2}} \qquad (6)$$

where $a = M_{20} / M_{00} - (M_{10} / M_{00})^2$,
$b = 2(M_{11} / M_{00} - M_{10} M_{01} / M_{00}^2)$,
and $c = M_{02} / M_{00} - (M_{01} / M_{00})^2$.

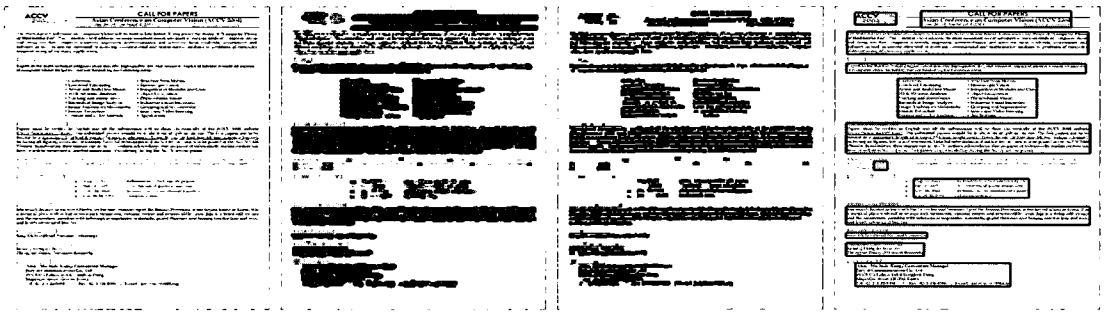
그림 10은 CAMShift 알고리즘 수행 중 검색창의 위치와 크기의 변화를 보인다. 그림 10에서 각 검색창의 위치와 크기가 변하면서 하나의 입력 영상에서의 문자열이 검출되는 과정을 볼 수 있다. 검은색 부분은 텍스춰 분석이 수행되지 않은 영역들이다.

이러한 CAMShift 방법으로 기존의 텍스춰 기반의 방법들에서 문제점으로 지적되는 입력 영상 전역 탐색에 관한 문제점을 어느 정도 해결할 수 있다. 그림 11은 비디오 영상에 대한 문자검출 결과의 중간 단계를 보인다.

(a) (b) (c) (d) (e)

그림 12. 컬러 문서 영상에 대한 문자 추출 예

Fig. 12. Localization examples for color document images.



(a) (b) (c) (d)

그림 13. 문서 영상 결과: (a)입력 영상, (b)신경망 출력, (c)연결 성분 분석 필터링, (d)문자 영역

Fig. 13. Intermediate result for document images: (a)input image, (b)TPI, (c)CC-based filtering, and (d)final result.

표 1. 영상 데이터베이스

Table 1. Experimental database.

Types	Scanned Images	Video Frames	MoCA samples	Web. Images
Number of Images	50	500	30	50
Image Sizes (pixels)	500□700	320□240	355□288~ 384□288	80□40~ 410□328
Text Sizes (pixels)	6 ~ 62	7 ~ 47	5 ~ 42	8 ~ 50

V. 실험 및 결과

5.1 데이터베이스

캡춰된 비디오 영상, MoCA 프로젝트의 실험 영상[1], 스캔 영상, 웹 영상 등 다양한 실험 데이터를 이용하여 제안된 방법을 테스트하였다 (표 1). MLP의 학습을 위하여 비디오 영상에서 50개의 프레임 (57000개의 학습 패턴)을 사용하고, 나머지 프레임은 테스트에 사용하였다. 오류 검출의 편의를 위하여, 학습 영상과 테스트 영상 내의 모든 문자 영역을 사각형으로 표시하여 각 좌표를 수작업으로 구하였다.

많은 연구 결과에서 문자 추출의 결과를 비교하기 위해, OCR 단계, 문자(character) 단계, 화소(pixel) 단계 등의 여러 단계에서 정확도를 측정하고 있는데[15, 22], OCR 단계에서의 성능 비교는 문자 인식기의 성능에 영향을 받을 수 있고, 문자 단계는 수작업으로 문자의 개수를 세는 작업이 힘들기 때문에, 본 연구에서는 화소 단계에서 정확도(precision)와 검출률(recall rate)을 이용하여 측정하였다.

$$\textit{precision}\ (\%) = \frac{\#\ of\ correctly\ \ detected\ \ text\ \ pixels}{\#\ of\ \ detected\ \ text\ \ pixels} \times 100 \tag{7}$$

$$\textit{recall}\ (\%) = \frac{\#\ of\ correctly\ \ detected\ \ text\ \ pixels}{\#\ of\ \ text\ \ pixels} \times 100 \tag{8}$$

5.2 문자 추출 예

그레이 문서 영상과 컬러 문서 영상에 대해서 제안한 방법을 실험하였다. 스캔받은 문서 영상에 대해서는 보다 깨끗한 문자 추출 결과를 위해서 X-Y 재귀 분할 알고리즘을 사용하였다. 그림 12는 다양한 문서 영상에 대한 문자 추출 결과를 보인다. 그림 12의 (b)와 (e)에

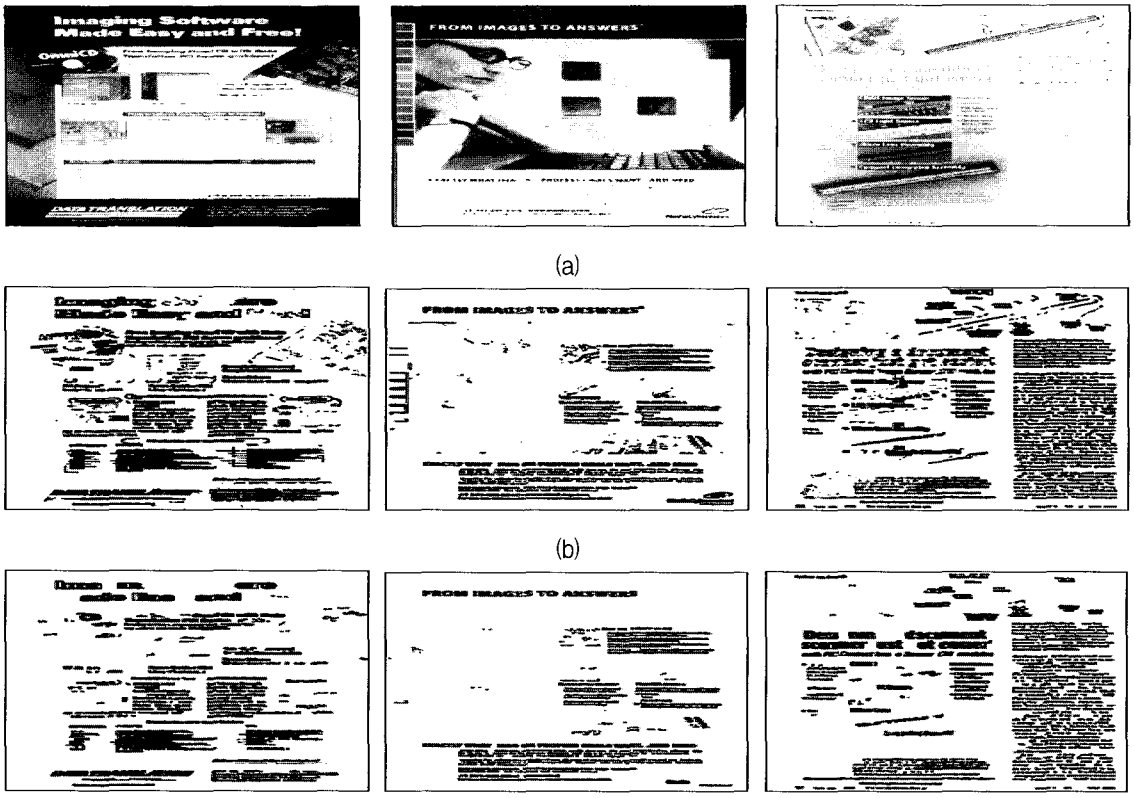


그림 14. 연결 성분 분석 예: (a) 입력 영상, (b)연결 성분 분석 전, (c) 연결 성분 분석 후
Fig. 14. CC analysis: (a) input images, (b) before, and (c) after CC-based filtering.

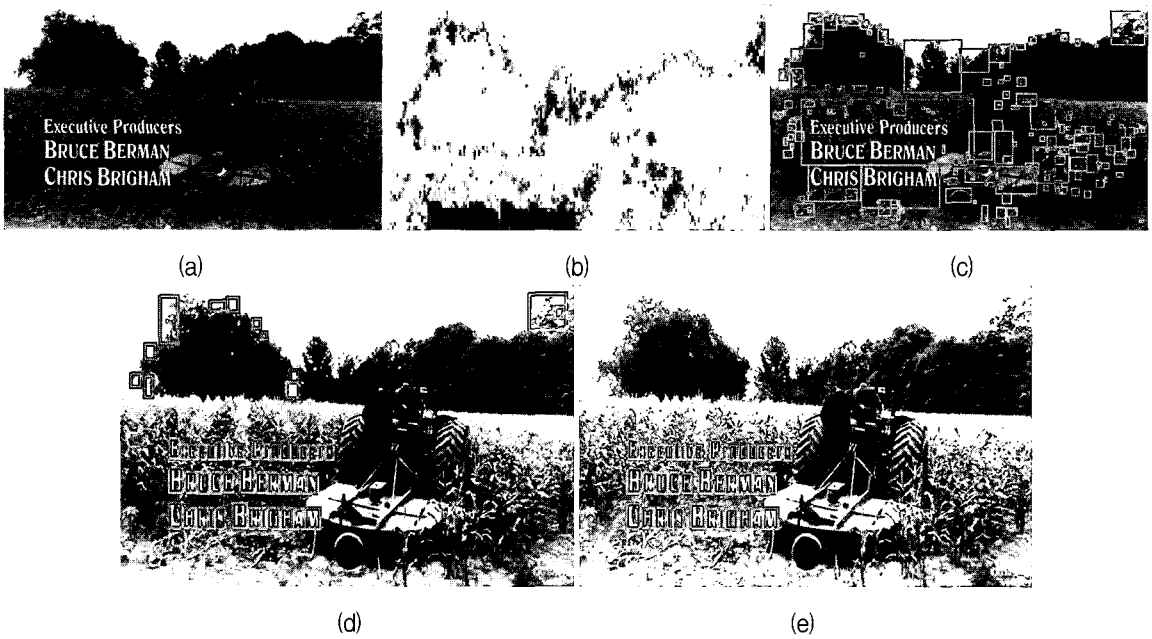


그림 15. 비디오 영상:(a)입력 영상, (b)텍스춰 분석 후, (c)연결 성분, (d) Stage 1, 2 필터링, (e) NMF 필터링
Fig. 15. Video images: (a)input image, (b)texture analysis, (c)CCs, (d)CC-based filtering, and (e)NMF-based filtering.

서 몇몇 글자를 검출하지 못하였는데, 이는 글자의 크기가 커서 필터링되거나 배경과의 색상변화가 적어서 MLP가 검출하지 못한 것이다. 그림 13은 그레이 문서 영상에 대한 문자 추출 예를 보이는 그림이다. (a)는 입력 영상, (b)는 신경망의 출력, (c)는 연결 성분 분석을 통한 필터링 결과, (d)는 최종 문자 영역 마킹 결과이다. (b) 그림에서 흰색 배경의 연한 회색의 문자들과 회색 배경의 검은색 글자들은 검출되지 않았다. 그림 14는 연결 성

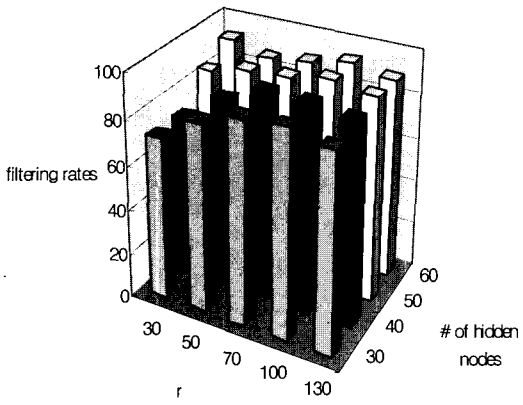


그림 16. 필터링 비율
Fig. 16. Filtering rates v.s. number of hidden nodes and r.

표 2. 비디오영상에 대한 정확도와 검출률(320□240 크기)
Table 2. Comparison of precision and recall for video images.

방법		Proposed Method	MLP + Histogram analysis	Connected Component
시간(sec.)		0.6	1.7	0.4
화소 단위(%)	정확도	96.1	87.2	78.8
	검출률	97.2	89.3	84.5

표 3. 문서영상에 대한 정확도와 검출률 비교
Table 3. Comparison of precision and recall for document images.

방법		Proposed Method	MLP + Histogram analysis	Connected Component
시간(sec.)		6.7	4.7	1.6
화소 단위(%)	정확도	94.1	92.2	88.8
	검출률	98.4	93.1	91.5

분을 이용한 필터링 전(b)과 후(c)의 비교 예이다. 그림 15는 비디오 영상에 대한 문자 추출 예이며, 연결 성분 필터링을 보이기 위해 CAMShift를 적용하지 않았다.

5.3 결과 분석

그림 16은 연결 성분 분석을 통한 필터링의 비율을 측정한 그림이다. 500□700 크기의 문서 영상 10장에 대해서 분석한 결과, 총 연결 성분 중 오추출에 해당하는 연결 성분들의 필터링 비율을 나타낸 그림이다. 실험에서 보이는 바와 같이 필터링 신경망의 입력 노드는 100개, 하나의 은닉층에 대해 은닉 노드의 개수가 50, 출력노드는 2개인 신경망을 사용하여, NMF의 파라미터 r이 100인 경우에 필터링 비율이 가장 높았다.

표 2와 3은 각각 비디오 영상과 문서 영상에 대한 연결 성분 방법[6]과 전체 영역을 탐색하는 MLP 방법[8]과 비교한 결과이다. 수행시간과 성능면에서 기존의 전체 영역 탐색 방법과 연결 성분 방법에 비해 장점을 보인다. 제한한 방법은 비디오 영상에 대해서 연결성분 기반 방법과 비슷한 수행시간에 상대적으로 높은 검출률과 정확도를 보이는데, 전체영역 탐색 방법에 비해 수행 속도가 3배 정도 향상되었으며 추출률 또한 향상된 것을 알 수 있다. 또한 문서 영상에 대해서는 전체 영역 탐색 방법과 유사한 수행시간에 향상된 추출률을 보인다.

VI. 결 론

본 논문에서는 텍스춰 방법과 연결 성분 방법을 입력 영상에 적합하게 결합한 문자 추출 방법을 제안하였다. MLP를 이용하여 다양한 환경의 영상에 대한 텍스춰 분석기를 별도의 특징값 추출없이 자동으로 구현하였으며, 연결 성분 분석을 통하여 텍스춰 방법의 국부성을 극복하며 비-문자 성분을 효과적으로 필터링하며 문자를 배경에서 분리할 수 있었다. 텍스춰 방법으로 검출률을 높이며 연결 성분 방법으로 정확도를 향상시켰다. 또한 CAMShift 알고리즘을 문자 검출에 사용함으로써, 전체 영상을 탐색하지 않고도 빠른 시간에 더욱 정확한 문자 영역을 구할 수 있었다. 향후의 연구 과제로 압축된 동영상에서의 수행 속도 개선을 위한 문자 추적, 더욱 자연스러운 모양의 문자열의 추출, 문자 인식 시스템과의 연계, 입력 영상의 자동 분류를 통한 효과적 시스템 구축에 관한 연구도 필요하다.

참 고 문 헌

[1] Rainer Lienhart and Frank Stuber, "Automatic Text Recognition In Digital Videos," *SPIE-The International Society for Optical Engineering*, pp. 180-188, 1996.

[2] Hae-Kwang Kim, "Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database," *Journal of visual communication and image representation*, Vol. 7, No. 4, December, pp. 336-344, 1996.

[3] Huiping Li, David Doerman, and Omid Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Transactions on Image*

Processing, Vol. 9, No. 1, pp.147-156, 2000.

[4] Yu Zhong, Hongjiang Zhang, and Anil K. Jain, "Automatic Caption Localization in Compressed Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, 2000.

[5] Anil. K. Jain, and Bin Yu, "Automatic Text Location in Images and Video Frames," *Pattern Recognition*, Vol. 31, No. 12, pp.2055-2076, 1998.

[6] E.Y. Kim, K.Jung, K.Y.Jeong, and H.J.Kim, "Automatic Text Region Extraction Using Cluster-based Templates," *International Conferen -ce on Advances in Pattern Recognition and Digital Techniques*, pp. 418-421, 2000.

[7] Yu Zhong, Kalle Karu, and Anil K. Jain, "Locating Text In Complex Color Images," *Pattern Recognition*, Vol. 28, No. 10, pp. 1523-1535, 1995.

[8] K. Y. Jeong, K. Jung, E. Y. Kim, and H. J. Kim, "Neural Network-based Text Location for News Video Indexing," *Proceedings of Interna -tional Conference of Image Processing*, 1999.

[9] Yassin M. Y. Hasan and Lina J. Karam, "Morphological Text Extraction from Images," *IEEE Transactions on Image Processing*, Vol. 9, No. 11, pp. 1978-1983, 2000.

[10] S. Messelodi and C. M. Modena, "Automatic Identifacation and Skew Estimation of Text Lines in Real Scene Images," *Pattern Recogni -tion*, Vol. 32, pp. 791-810, 1999.

[11] Victor Wu, Raghavan Manmatha, and Edward M. Riseman, "TextFinder: An Automatic System to Detect and Recognize Text in Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21. No. 11, pp. 1224-1229, 1999.

[12] C. Strouthopoulos and N.Papamarkos, "Text Identification For Document Image Analysis Using a Neural Network," *Image and Vision Computing*, Vol. 16, pp. 879-896, 1998.

[13] Keechul Jung, "Neural Network-based Text Location using Color Texture Discrimination," *PhD. Thesis*, Artificial Intelligence Laboratory, Kyungpook National University, Korea, December 1999.

[14] Huiping Li and David Doermann, "A Video Text Detect System based on Automated Training," *International Conference on Pattern Recognition*, pp.223-226, 2000.

[15] Axel Wernicle and Rainer Lienhart, "On the Segmentation of Text inVideos," *IEEE Inter -national Conference on Multimedia and Expo*, Vol. 3, pp. 1511-1514, 2000.

[16] Ullas Gargi, Sameer Antani, and Rangachar Kasturi, "Indexing Text Events in Digital Video Database," *International Conference on Pattern Recognition*, pp. 1481-1483, 1998.

[17] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39-51, 1998.

[18] Anil K. Jain and Kalle Karu, "Learning Texture Discrimination Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 2, pp. 195-205, 1996.

[19] Yizong Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, August, pp.790-799, 1995.

[20] Gary R. Bradski and Vadim Pisarevsky, "Intel's Computer Vision Library: Application in Calibration, Stereo, Segmentation, Tracking, Gesture, Face and Object Recognition," *Proceedings of IEEE Conference of Computer Vision and Pattern Recognition*, Vol. 2, pp. 796 797, 2000.

[21] Dorin Comaniciu and Visvanathan Ramesh, "Robust Detection and Tracking of Human Faces with an Active Camera," *The 3rd IEEE International Workshop on Visual Surveillance*, pp.11-18, 2000.

[22] Sameer Antani, Ullas Gargi, David Crandall, Tarak Gandhi, and Rangachar Kasturi, "Extraction of Text in Video," *Technocal Report*, CSE-99-016, August 30, 1999.

[23] B.K.P. Horn, Robot Vision. MIT Press, 1986.

[24] Rainer Lienhart and Frank Stuber, "Automatic Text Recognition In Digital Videos," *SPIE-The International Society for Optical Engineering*, pp. 180-188, 1996.

[25] Hae-Kwang Kim, "Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database," *Journal of Visual Communication and Image Representati -on*, Vol. 7, No. 4, December, pp. 336-344, 1996.

[26] Huiping Li, David Doerman, and Omid Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Transactions on Image Processing*, Vol. 9, No. 1, January, pp.147-156, 2000.

[27] Yu Zhong, Hongjiang Zhang, and Anil K. Jain, "Automatic Caption Localization in Compressed Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, pp. 385-392, 2000.

[28] Anil. K. Jain and Bin Yu, "Automatic Text

Location in Images and Video Frames," *Pattern Recognition*, Vol. 31, No. 12, pp.2055-2076, 1998.
[29] E.Y. Kim, K. Jung, K.Y. Jeong, and H.J. Kim, "Automatic Text Region Extraction Using Cluster-based Templates," *International Conference on Advances in Pattern Recognition and Digital Techniques*, pp. 418-421, 2000.
[30] K. Jung, "Neural Network-based Text Location in Color Images," *Pattern Recognition Letters*, Vol. 22, No. 14, pp. 1503-1515, 2001.
[31] 정기철, 김광인, 한정현, "신경망 기반의 텍스춰 분석을 이용한 효율적인 문자 추출", *정보과학회 논문지*, Vol. 29, No. 3, pp. 180-191, 2002.
[32] D. D. Lee, H. S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature* 401, pp. 788-791, 1999.
[33] H. S. Seung, "Derivation of the objective function (Eq.2)," http://jounalclub.mit.edu.
[34] D. D. Lee, H. S. Seung, "Algorithms for non-negative matrix factorization," *In Advances in Neural Information Processing Systems*, 13, pp. 556562, 2001.
[35] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification," *Wiely-Interscience*, 2000.

저 자 소 개

정 기 철(정회원)-교신저자
1996년 경북대학교 컴퓨터공학과 공학석사
2000년 경북대학교 컴퓨터공학과 공학박사
1999년~2000년 Machine Understanding Division, ElectroTechnical Laboratory, Japan, 방문연구원
2001년~2002년 미국 미시간대 Anil K. Jain 교수 PRIP 연구실 박사 후 연구원,
2003년~현재 숭실대학교 정보과학대학 미디어학부 교수
<주관심분야: Interactive Contents, HCI, 영상 처리, 패턴 인식, Augmented Reality, Mobile Vision System, 인공지능>