



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

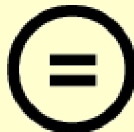
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

다층 탐색을 이용한 컨테이너
식별자 영상 인식 시스템



2012년 2월

부경대학교 산업대학원

컴 퓨 터 공 학 과

안 성 윤

공 학 석 사 학 위 논 문

다층 탐색을 이용한 컨테이너
식별자 영상 인식 시스템



부경대학교 산업대학원

컴 퓨 터 공 학 과

안 성 윤

이 논문을 안성윤의 공학석사
학위논문으로 인준함

2012년 1월 5일



주심 공학박사 정 순 호 (인)

위원 공학박사 서 경 룡 (인)

위원 공학박사 우 종 호 (인)

목 차

목차	i
요약	vi
Abstract	viii
제 1 장 서론	1
제 2 장 컨테이너 식별자 영상 인식 전처리	3
2.1 컨테이너 식별자의 특징 및 배열 구조	3
2.2 컨테이너 식별자 인식을 위한 영상의 이진화	5
2.2.1 영상변환	5
2.2.2 이진화(Binarization)	7
2.2.3 레이블링(Labeling)	16
2.2.4 패넬링(Paneling)	17
2.2.5 패넬병합(Panel Merge)	19
2.3 개별문자 분리	20
2.3.1 문자영역의 이진화	21
2.3.2 문자영역의 기울기 보정	23
2.3.3 분리(Segmentation)	24
제 3 장 컨테이너 식별자 영상 인식 및 시스템	26
3.1 개별 문자의 인식을 위한 신경망 설계	26
3.1.1 다층 퍼셉트론	26
3.1.2 신경망 설계	27
3.1.3 신경망 데이터 설계	30
3.1.4 정규화	34

3.1.5 문자인식	34
제 4 장 실험 결과 및 고찰	36
4.1 실험 환경	36
4.2 신경망의 학습 정보	38
4.3 컨테이너 번호 인식 실험	39
제 5 장 결론	42
참고문헌	43



그 립 목 차

[그림 1] 컨테이너 번호 시스템	4
[그림 2] 외부 환경의 노출과 굴곡 특징	4
[그림 3] 컨테이너 식별자의 배열구조	5
[그림 4] RGB 색상 모델	6
[그림 5] 영상변환의 예	7
[그림 6] 히스토그램의 예	8
[그림 7] 히스토그램 스무딩	8
[그림 8] 흑백영상의 3차원 구성 방법	9
[그림 9] 3차원 분석의 z축 누적그래프와 히스토그램의 일치	10
[그림 10] 문자의 밝기가 배경의 밝기에 포함되는 경우	11
[그림 11] 그림 10의 z축 누적그래프와 이진화 결과	11
[그림 12] 문자의 밝기가 배경보다 어두운 경우	12
[그림 13] 그림 12의 z축 누적그래프와 이진화 결과	12
[그림 14] 문자의 밝기가 배경보다 밝은 경우	13
[그림 15] 그림 14의 z축 누적그래프와 이진화 결과	13
[그림 16] 다층탐색기의 예	14
[그림 17] 다층탐색의 임계치를 적용한 이진화한 영상	15
[그림 18] 레이블링의 예	16
[그림 19] 레이블링 결과	17
[그림 20] 군집상태 분석과 패넬링	18
[그림 21] 패넬링 결과	18
[그림 22] 행간 패넬병합	19

[그림 23] 층간 패널병합	20
[그림 24] 개별문자 분리 과정	21
[그림 25] 양봉형태	21
[그림 26] 다봉형태	22
[그림 27] 단봉형태와 임계치 찾기	22
[그림 28] 레이블링을 이용한 기울기 보정 알고리즘의 결과	23
[그림 29] 한 문자가 여러 조각으로 분리된 상태에서의 처리	24
[그림 30] 손상된 상태에서의 처리	25
[그림 31] 문자가 붙어있는 상태에서의 처리	25
[그림 32] 누적그래프 분석을 잡음 제거에 활용	25
[그림 33] 다층 퍼셉트론의 구조	26
[그림 34] 뉴런의 설계	27
[그림 35] 층을 표현하는 설계	28
[그림 36] 각층에서 수상돌기와 축색돌기의 구성	28
[그림 37] 신경망 클래스	29
[그림 38] 신경망 데이터 설계	30
[그림 39] 학습 프로그램	32
[그림 40] 패턴 폴더	33
[그림 41] 정규화의 예	34
[그림 42] 신경망의 입력패턴과 출력패턴	35
[그림 43] 실험에 사용한 분석 프로그램	36
[그림 44] 실험에 사용한 결과 프로그램	37
[그림 45] 학습폰트 작성기	38
[그림 46] 학습폰트 작성기 결과의 일부	38
[그림 47] 인공신경망의 학습 정보	39

[그림 48] 육안인식이 가능하나 실패한 경우	40
[그림 49] 육안인식이 불가능한 경우	41

표 목 차

[표 1] 가중치테이블의 예	31
[표 2] 결과테이블의 예	32
[표 3] 실험 결과 인식률	40

다층 탐색을 이용한 컨테이너 식별자 영상 인식 시스템

안 성 윤

부 경 대 학 교 산 업 대 학 원 컴 퓨 터 공 학 과



요 약

카메라에서 입력받은 컬러 영상을 흑백으로 변환하여 영상에서 밝기에 따라 군집을 이루는 영역을 층으로 설정했다. 각 평면의 누적개수를 분석하는 다층탐색으로 이진화의 임계치를 찾아 영상을 이진화했다. 레이블링을 사용해서 문자 후보를 선별하고 패넬링을 통해 문자영역 후보를 생성했다. 패넬병합은 행간병합, 열간병합, 그리고 층간병합으로 수행해서 문자영역의 후보를 줄인다. 문자영역에서 개별문자의 분리를 위해서 다시 히스토그램분석을 수행해서 이진화의 임계치를 구한다. 앞서 수행했던 다층탐색의 각 층에서의 정보를 활용하여 방향성을 결정하며 문자영역의 이진화 영상을 생성한다. 이 영상에서 레이블링을 수행하였고 레이블간의 위치정보를 분석하여 기울어짐을 보정했다. 이러한 전처리과정으로 정확한 문자영역의 검출과 개별문자의 분리율을 높여 인식률을 향상

시켰다. 분리한 개별문자의 인식에 인공신경망을 사용하였다. 실험 결과 기존의 상용제품에서 인식에서 실패한 영상들에 대해서도 80%를 인식할 수 있었다. 직접 촬영한 영상에서는 100%의 인식률을 보였다.

향후 부적절한 조명, 문자의 멸실 등 다양한 환경에서의 영상들을 대상으로 인식률을 향상시키는 연구가 필요하다.



Recognition System for Container Identifier Image Using Multilayer Searching

An Seong-Yun

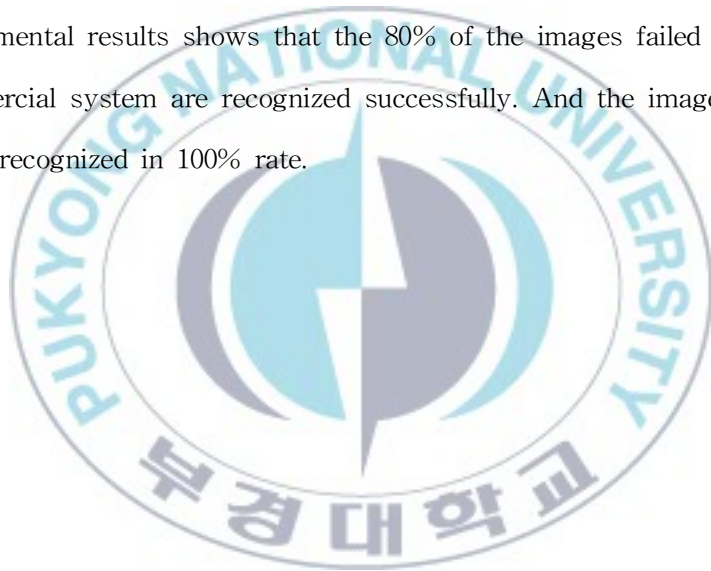
*Department of Computer Engineering, Industrial Graduate
School,*

Pukyong National University

Abstract

The color image captured from the camera is transformed to the black and white image and is got the histogram from the image. The layers formed the groups according to the brightness of picture elements are set and we can get the threshold value for the image binarization through analyzing the cumulated elements of these layers in the histogram. The character candidates for the identifier are selected by using labeling the image and the characters regions are set up by using image panning. The number of the characters region candidates can be reduced by merging the panels such as the rows merging, the columns merging, and layers merging. In other to segment

the each character from the acquired characters region, the histogram of this image is analyzed again and the threshold value is got for binarization. The binary image of characters region is created by utilizing the information taken from the multilayer searching. In this binary image the labeling is conducted and the leaned characters are adjusted by analyzing the locations of the labels. Through the procedure of preprocessing the image we can get exactly the character region and improve the character segmentation and the recognition rate. The segmented characters are recognized in the neural network. The experimental results shows that the 80% of the images failed in the existed commercial system are recognized successfully. And the images taken directly are recognized in 100% rate.



In future the studies on the images taken in various environments such as the improper illumination and including the damaged characters are required.



1. 서론

효율적인 항만 운영을 위해 많은 자동화·무인화 기술들을 개발하거나 적용하고 있으며 그 중 컨테이너의 식별은 게이트, 야드, 본선에서 사용하는 주요기술로써 작업의 효율을 증대시키고 비용을 감축시키는 역할을 한다. 컨테이너는 다른 컨테이너와 구분할 수 있는 고유 식별자를 가지고 있다[1]. 컨테이너를 반입, 반출 및 처리하는 과정에서 식별자를 확인하는 작업이 선행되어야 한다. 아직 대부분의 항만에서는 식별자의 확인이 작업자의 개입으로 이루어진다. 이는 근무자의 상태에 따라 오류를 발생시킬 수 있고, 시간 및 비용 면에서 비효율적이다. 따라서 이를 자동화함으로써 인건비를 줄이고, 컨테이너의 체류시간을 최소화 하며, 반입·반출되는 컨테이너에 대한 정확한 정보관리가 가능하게 될 것이다. 이러한 자동화를 위해서는 컨테이너 번호판에 대한 영상 처리 기술이 필요하다[2]. 영상처리를 기반으로 하는 식별자 인식 시스템은 문자영역검출과 개별 문자분리 그리고 인식 단계로 구성된다. 이 중에서 문자영역을 검출하는 단계가 핵심 기술이며 여러 연구가 진행되었다. Prewitt, Canny 등의 마스크로 에지를 검출하여 형태학적 특징 정보나 누적그래프를 이용한 방법과 모폴로지 기법을 응용한 Top-Hat Transform을 이용한 방법으로 문자영역을 검출하였다[2-4].

컨테이너 식별자 영상 인식 시스템은 일반적으로 센서는 컨테이너 차량이 진입함에 따라 근접 트리거(trigger) 센서가 작동하여 진입한 차량의 위치에 따라 필요한 부분의 영상을 획득하기 위한 트리거 신호로 사용된다. 수대의 카메라가 동시에 작동하여 운송차량의 컨테이너에 표시된 식별자 번호를 촬영한다. 인식 시스템의 이미지 그래버(image grabber) 보드에 전송되어 인식시스템에서 전처리 과정과 인식 알고리즘을 수행한다. 이 때 근접 센서 신호와 카메라 영상은 적절한 케이블로 전송된다.

본 연구에서는 컨테이너 식별자 영상 인식 시스템을 구현한다. 영상 인식 시스템은 전처리와 문자인식의 단계로 구성되며, 전처리는 문자영역검출과 개별문자분리단계로 구분된다. 문자영역검출에서는 이진화의 임계치를 찾기 위해 흑백영상을 x, y 평면으로 설정하고, 영상의 밝기를 z 축으로 하는 3차원 영상을 구성한다. 여기서 z 축 평면의 누적그래프를 구하고 수개의 지점을 임계치로 하는 다층탐색으로 이진화 후보영상들을 생성한다. 이 영상들은 레이블링과 패넬링을 수행하고 패넬병합 과정을 거쳐 문자영역을 검출한다. 개별문자분리는 구한 문자영역에서 잡음을 제거하고 배경과 문자를 분리하는 과정이다. 문자영역에서 레이블링을 수행하고 레이블들의 중심관계를 분석하여 영상의 기울어짐을 보정한다. 또한 레이블링과 누적그래프의 병행분석으로 개별문자 분리와 잡음제거에 사용한다. 문자인식은 인공신경망의 오류역전파알고리즘을 사용하였다. 컨테이너 번호는 영문자와 숫자로 조합되어있는데 하나의 신경망으로도 가능하지만 숫자와 영문에 비슷한 문자의 존재와 새로운 패턴이 추가되었을 때, 빠른 학습과 좀 더 명확한 분류를 위해 영문과 숫자로 분리하여 인식시켰다.

실제 실험에는 해외인식기에서 실패한 측면 영상 105장, 직접 촬영한 측면 영상 38장, 직접 촬영한 앞면과 뒷면 19장 등 총 162개의 샘플영상을 사용하였다. 실험 결과 상용제품에서는 실패한 영상에서 80%이고, 육안으로 인식 가능한 영상의 인식률은 91.3%를 보였다. 직접 촬영한 영상에서는 100%의 인식률을 보였다.

2. 컨테이너 식별자 영상 인식의 전처리

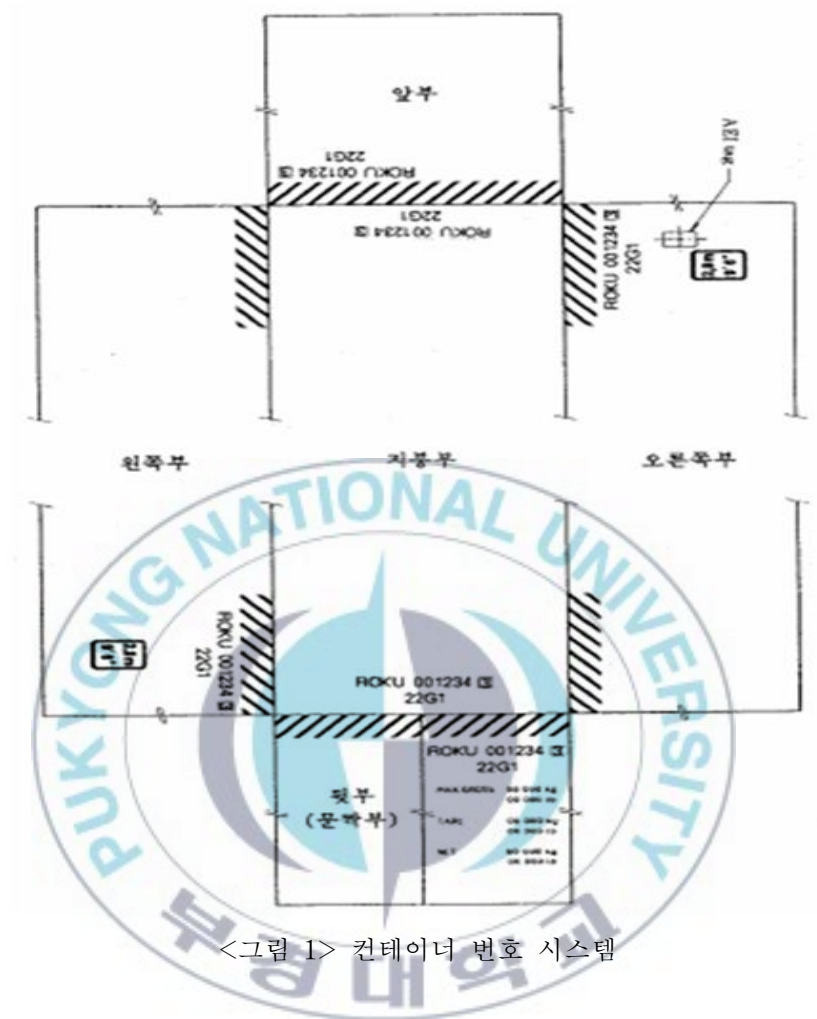
컨테이너 식별자 영상 인식은 전처리와 인식단계로 구성된다. 전처리는 문자영역을 검출하고 개별문자를 분리하는 과정이며, 인식은 분리된 문자를 정규화시키고 인공신경망으로 문자를 인식시킨다. 문자영역검출은 이진화와 레이블링 수행하고 패널링을 통해 문자영역을 확보하고 패널병합을 통해 문자영역의 후보를 추출한다. 개별문자분리는 이진화, 기울기보정, 잡음제거 및 분리 단계로 구성된다. 컨테이너 식별자 영상 인식의 전처리는 인식률을 결정짓는 중요한 과정이다.

2.1 컨테이너 식별자의 특징 및 배열 구조

국제 화물 컨테이너의 식별에 관한 기호는 소유자기호, 장비종류 식별기호, 일련번호, 체크디지트(check digit)로 구성되며, 소유자기호는 3개의 영문자로 구성된다. 장비종류 식별기호에 컨테이너는 U를 사용한다. 일련번호는 6자리 숫자로 구성되며, 체크디지트가 1개가 추가되어 일련번호의 정확성을 확인한다[1].

컨테이너의 번호는 그림 1과 같이, 바닥을 제외한 5면에 6개가 있다. 지붕부에는 상부번호와 하부번호 2개가 있지만, 지붕에는 운송 및 적재과정에서 손상이 있고 먼지 등의 이물질이 쌓여 오염이 심한 경향이 있으므로 일반적으로 영상인식에서는 제외한다. 문짝부는 문짝의 손잡이로 인해 오염이나 외부충격에 안전하기 때문에 가정 선명한 식별자 영상을 확보할 수 있는 특징이 있다.

컨테이너는 외부 환경에 노출되어 있기 때문에 기후 및 주야간, 외부 오염요소, 훼손 등과 같은 요소로 인하여 문자를 인식하는데 어려움이 따른다. 또한 문자종류, 색상, 크기, 두께, 배열구조 등이 일정하게 규정되어 있지 않고 컨테이너의 표면은 굴곡이 되어있는 특징으로 인하여 영상의 획득 위치에 따라 명암이 차이 나고 왜곡 및 형태변형으로 인식하는데 많은 어려움이 따른다.



<그림 1> 컨테이너 번호 시스템



<그림 2> 외부 환경의 노출과 굴곡 특징

컨테이너 번호의 배열구조 또한 규약이 없기 때문에 어려움이 따른다. 가장 많이 나타나는 구조는 1행과 1열이며, 그 다음으로 2행과 4행을 볼 수 있다. 3행과 2열 구조도 간혹 나타난다.



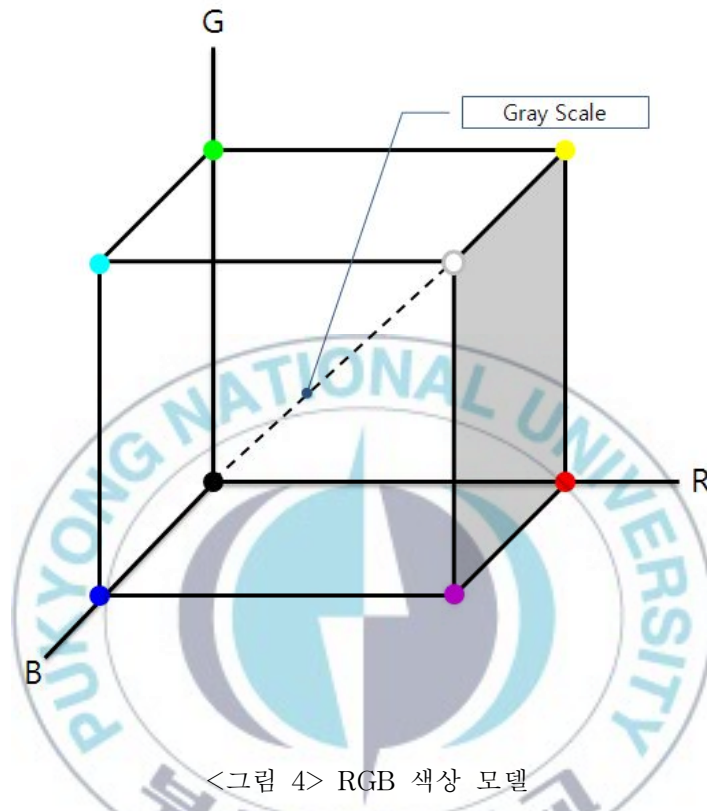
<그림 3> 컨테이너 식별자의 배열구조

2.2 컨테이너 식별자 인식을 위한 영상의 이진화

2.2.1 영상변환

획득한 영상은 30만화소(640*480) RGB타입의 트루컬러영상이다. 컨테이너 식별자는 색상에 대한 규정사항이 없기 때문에 컬러영상에서 특징을 찾을 수 없다. 그리고 3채널로 구성되기 때문에 1채널보다 3배의 처리시간이 필요하다. 이러한 특징으로 RGB영상은 흑백영상으로 변환한다. 흑백영상으로 변환하기 위해 그림 4와 같이 R, G, B 축으로 정의되는 단위 육면체를 이용하여 표현할 수 있다. 원점은 검정을 표현하고, 이와 대각선으로 반대편의 꼭지점이 흰색이다.

육면체의 각 축 상의 꼭지점은 원색을 표현하고, 나머지 꼭지점은 각 원색의 보색을 나타낸다.



RGB영상을 흑백으로 변환하는 가장 간단한 방법은 그림 4에서 원점과 흰색 꼭지점을 선으로 연결하는 직선처럼, 각 색상의 1/3을 반영하는 방법도 있으나, NTSC에서 표준으로 정하고 있는, 색상의 민감도에 따른 변환 방법은 식 1과 같다.

$$\text{Intensity} = 0.299R + 0.587G + 0.114B \quad (1)$$

그림 5는 식 1을 적용하여 RGB영상을 흑백영상으로 바꾼 예이다. 실험에 사용하는 모든 영상은 식 1의 공식을 이용하여 흑백영상으로 변환한다.

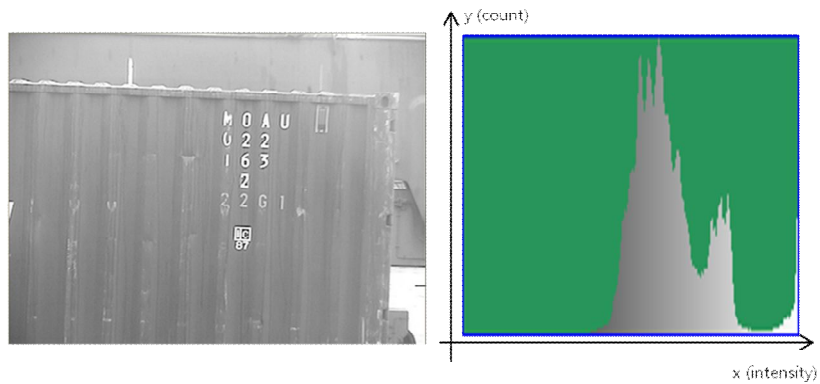


<그림 5> 영상변환의 예

2.2.2 이진화(Binarization)

흑백영상의 이진화에서 임계치(threshold)의 설정이 중요하며 이 값에 따라 인식률에 영향을 준다. 일반적으로 이진화의 임계치를 검출하는 방법으로 히스토그램(histogram) 분석을 사용한다. 히스토그램은 영상의 명암분포를 그래프형태로 보여주기 위해 사용되는 도구이다. 히스토그램은 영상의 명도 내용을 요약한 것이라 할 수 있으며 영상에 대한 많은 정보를 가지고 있고 계산하기 간편하다는 장점이 있다. 히스토그램에서 일반적으로 수평축은 밝기를 나타내고 수직축은 명암에 따른 누적개수를 표시한다.

그림 6은 흑백영상을 히스토그램으로 분석한 예이다. 영상에서 문자의 특징을 보면, 문자는 배경보다 밝으므로 히스토그램에서 255와 가까이 있는 봉이 문자 영역임을 직관적으로 알 수 있다. 그러나 원본 히스토그램은 밝기에 따라 누적되는 불연속그래프 형태로 나타나므로 피크(peak)와 밸리(valley)를 찾기 위해서 연속그래프 형태로 변환 시키는 히스토그램 스무딩(histogram smoothing)이 필요하다. 본 연구에서는 구하는 밝기의 지점을 포함하여 좌우로 수개의 범위에서 평균값을 취하는 평균필터(mean filter)로 스무딩하였다. 그림 6에 적용한 결과는 그림 7과 같다.



<그림 6> 히스토그램의 예

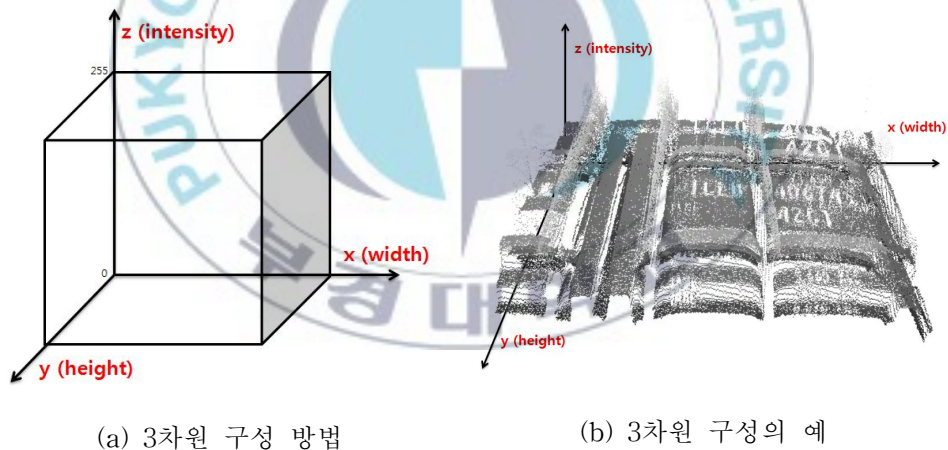


<그림 7> 히스토그램 스무딩

일반적으로 컨테이너 영상에서 문자의 밝기(intensity)는 배경의 밝기에 포함되거나, 배경보다 밝거나, 배경보다 어두운 특징이 있다. 이로 인해 한 개의 임계치를 사용하는 방법으로 다양한 영상에 적용하는 것은 한계가 따른다. 본 논

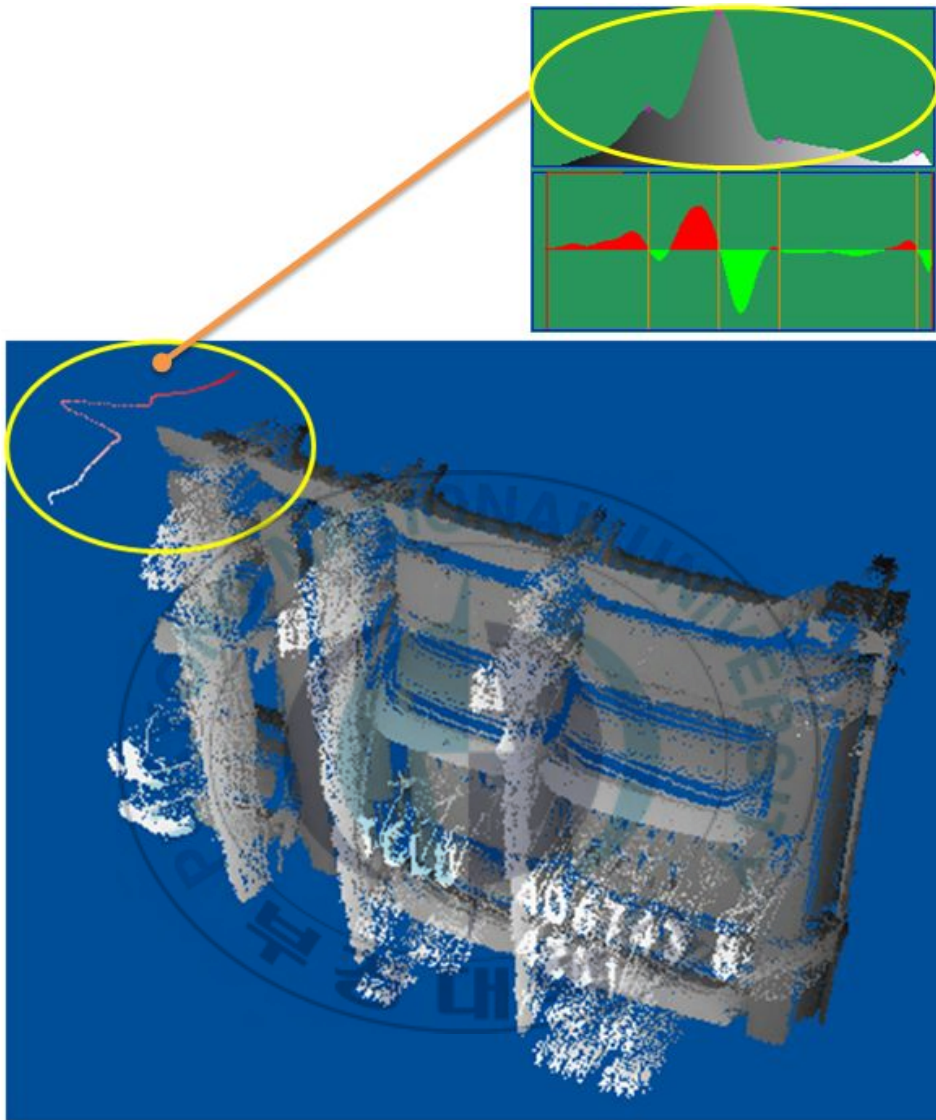
문에서는 히스토그램 분석을 통해 봉우리의 경계들을 임계치로 정하는 방법을 사용하였다. 이와 같은 방법은 봉우리의 개수가 변함에 따라 적용하는 임계치의 개수도 변하는 특징이 있다.

히스토그램에서 봉우리의 개수에 따라 임계치를 구하는 방법은 영상을 3차원으로 구성하면 구분의 판별이 용이해진다. 3차원 영상의 구성 방법은 흑백 영상을 x , y 평면으로 설정하고, 밝기를 z 축으로 한다. 여기서 z 축은 0에서 255의 범위를 가지며, 각 평면의 누적그래프는 히스토그램과 일치한다. 이와 같은 3차원 분석법은 배경과 문자의 위치와 히스토그램 관계를 시각적으로 잘 나타내고 임계치를 설정하는 방법에 관한 중요한 단서를 제공해준다. 그림 8의 (a)는 3차원 영상으로 구성하는 방법을 나타내고 (b)는 3차원 구성의 예이다.



<그림 8> 흑백영상의 3차원 구성 방법

3차원으로 구성한 영상의 z 축은 0에서 255의 범위를 가진다. 여기서 z 축 평면의 누적개수를 그래프로 변환해서 임계치를 결정하는 분석도구로 사용한다. 그림 9는 z 축의 누적그래프로 변환한 예이다.



<그림 9> 3차원 분석에서 사용하는 z축 누적그래프

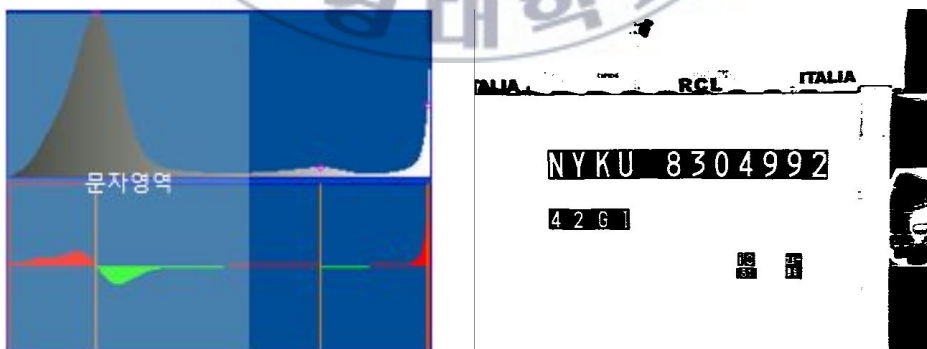
그림 10부터 15까지는 컨테이너의 배경과 식별문자 밝기에 따라 z축 누적그래프에서 문자의 위치를 결정하는 방법을 보여준다.

그림 10은 컨테이너 배경은 어둡고, 문자의 배경은 밝고 문자는 어두운 경우이다. 즉, 문자가 배경에 포함되는 것을 보여주고 있다.



<그림 10> 문자의 밝기가 배경의 밝기에 포함되는 경우

문자영역이 배경에 포함되어있는 영상의 히스토그램은 그림 11과 같이 가장 큰 봉이 배경영역이고, 배경영역에서 문자영역이 포함되어 있기 때문에 문자영역이 따로 드러나지 않으므로 배경영역에서 이진화 임계범위를 결정해야 한다.



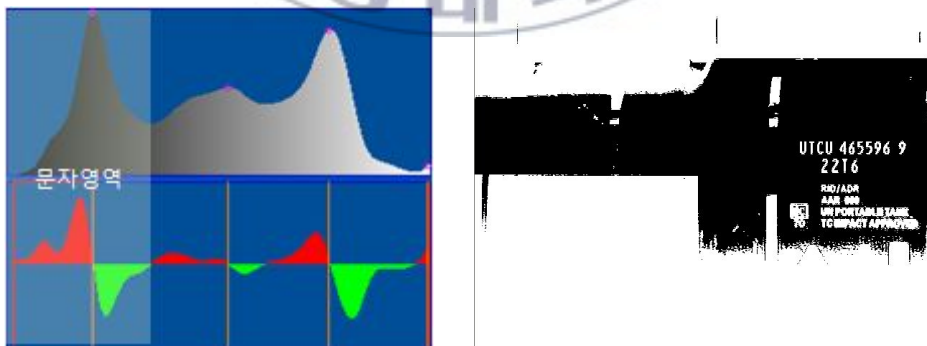
<그림 11> 그림 10의 z축 누적그래프와 이진화 결과

3차원 분석에서 z축은 표면이 255이기 때문에 어두워질수록 깊이 위치한다. 그림 12는 배경은 밝은 색이고 문자영역은 어둡기 때문에 배경보다 뒤에 위치하는 경우이다.



<그림 12> 문자의 밝기가 배경보다 어두운 경우

그림 13에서 가장 우측의 밝은 봉이 배경영역이고, 그 반대 쪽 어두운 부분이 문자영역이므로 이진화 임계범위는 어두운 부분에서 결정해야 한다.



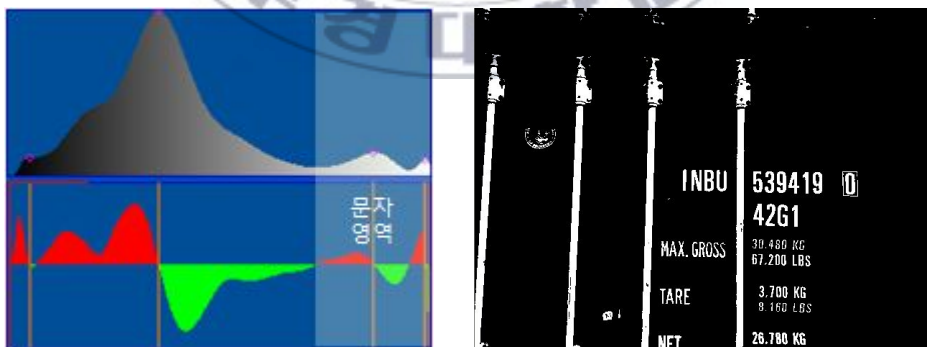
<그림 13> 그림 12의 z축 누적그래프와 이진화 결과

그림 14는 컨테이너의 배경의 밝기보다 문자영역이 더 밝은 영상이며, 문자가 배경보다 앞에 있어서 잘 드러난다.



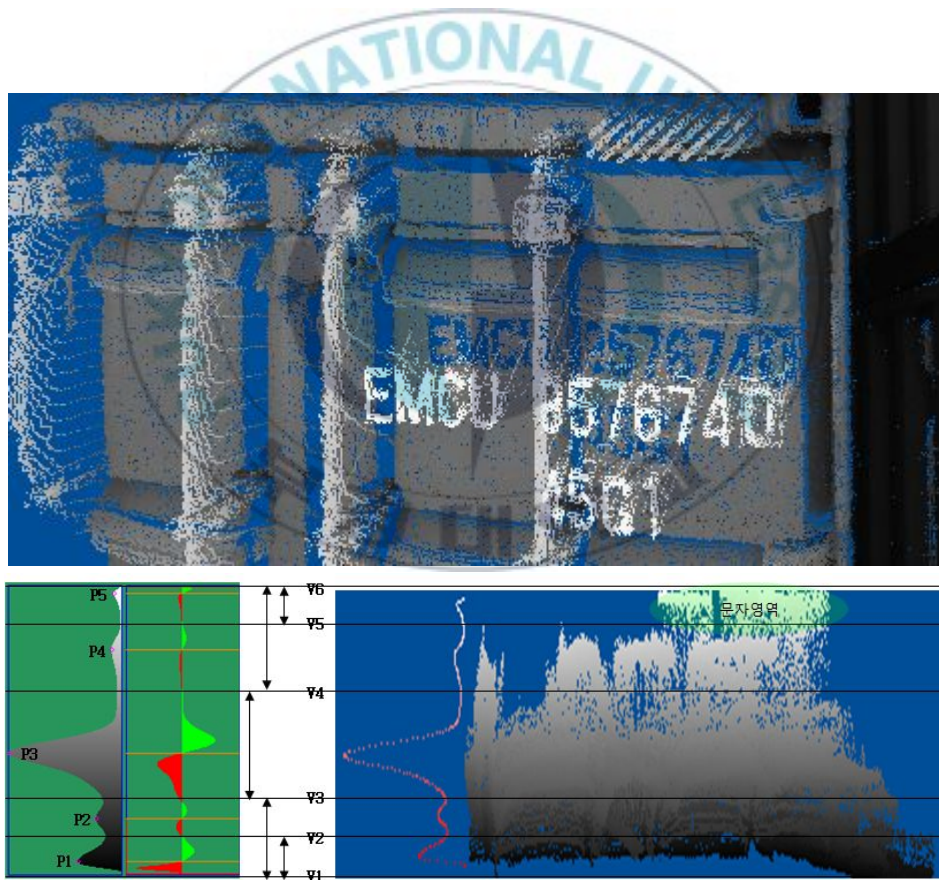
<그림 14> 문자의 밝기가 배경보다 밝은 경우

그림 15에서 우측의 밝은 부분이 문자영역이고, 어두운 부분이 배경영역이므로 이진화 임계범위는 밝은 부분에서 결정해야 한다.



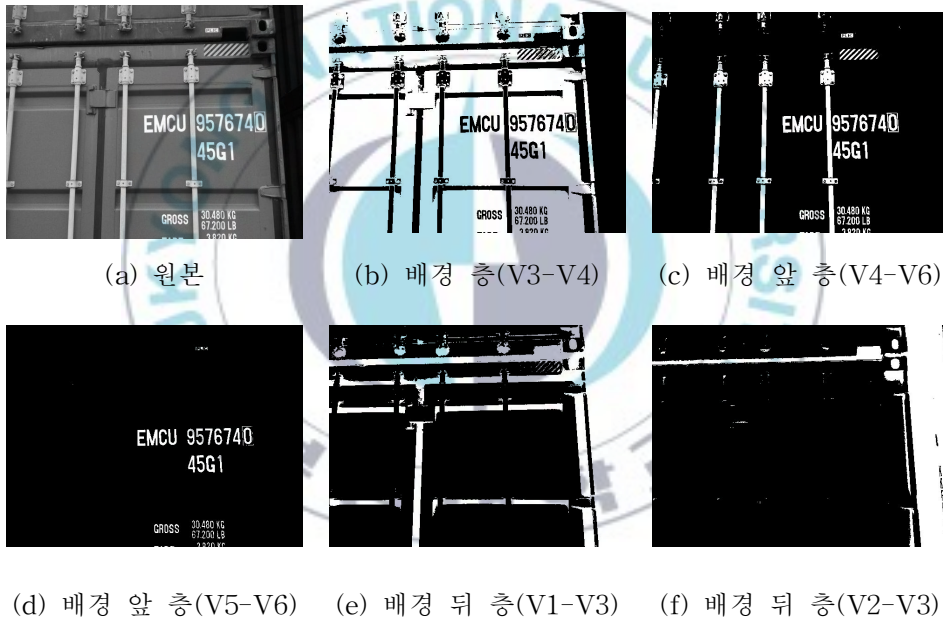
<그림 15> 그림 14의 z축 누적그래프와 이진화 결과

이러한 특징으로 3차원으로 분석한 컨테이너 영상에서 z축을 기준으로 자르면, 배경층과 배경의 앞층, 그리고 배경의 뒤 층으로 나누어지며 기본적으로 임계치는 3개가 된다. 배경의 앞 층이나 뒤 층 또한 누적그래프의 밸리에 의해 여러 층으로 나누어질 수 있으며, 임계치는 그 수만큼 늘어난다. 이와 같이 3차원 영상 z축 평면의 누적그래프에서 이진화 임계치를 정하는 방법을 다층탐색이라고 한다. 다층탐색으로 이진화 임계치를 정하는 기준은 그림 16에서 화살표에서 표시하고 있는 범위이며, 우측 그림은 3차원 분석 영상을 측면에서 봤을 때 문자영역을 표시하고 있다.



<그림 16> 다층탐색 예

그림 16에서 5개의 봉이 존재하며 V3-V4는 배경범위이다 먼저 배경영역인 V3-V4에 포함되는 범위에서 임계치를 구한다. V3은 임계치의 시작이고 V4는 임계치의 끝을 나타낸다. V3과 V4의 범위에 포함되면 1, 아니면 0으로 하는 이진화 영상을 생성할 때 사용된다. 그리고 배경의 앞 층의 임계치 범위 V4-V6과 배경의 뒤 층의 임계치 범위 V1-V3을 구한다. 그림 16에서는 앞 층은 2개의 봉으로 구성되기 때문에 임계치 범위 V5-V6이 추가로 생성되었다. 뒤 층 또한 마찬가지로 V1-V2가 생성되었다.

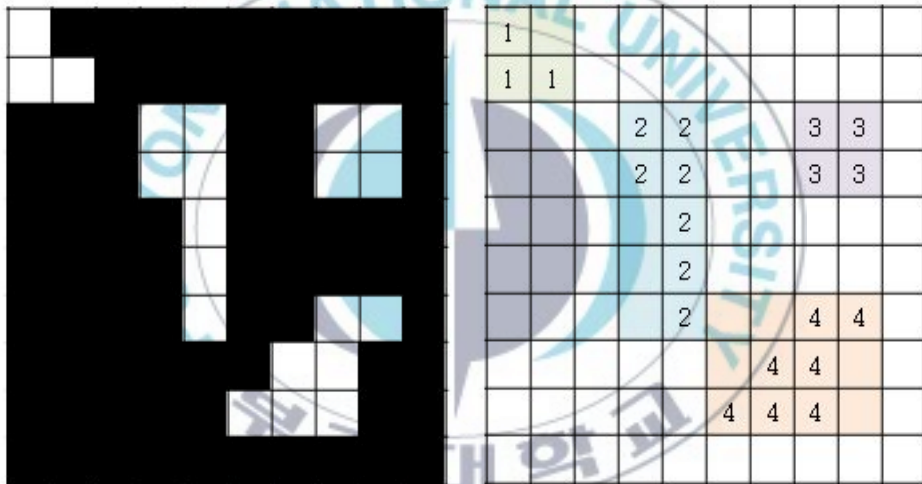


<그림 17> 다층탐색의 임계치를 적용한 이진화한 영상

그림 17은 다층탐색으로 생성한 임계치를 사용하여 이진화 한 결과이다. 5개의 층이 생성되었고 배경 앞 층에서 문자가 뚜렷이 보이는 결과를 볼 수 있다. 이와 같이 이진화는 다층탐색으로 수개의 후보영상을 생성한다.

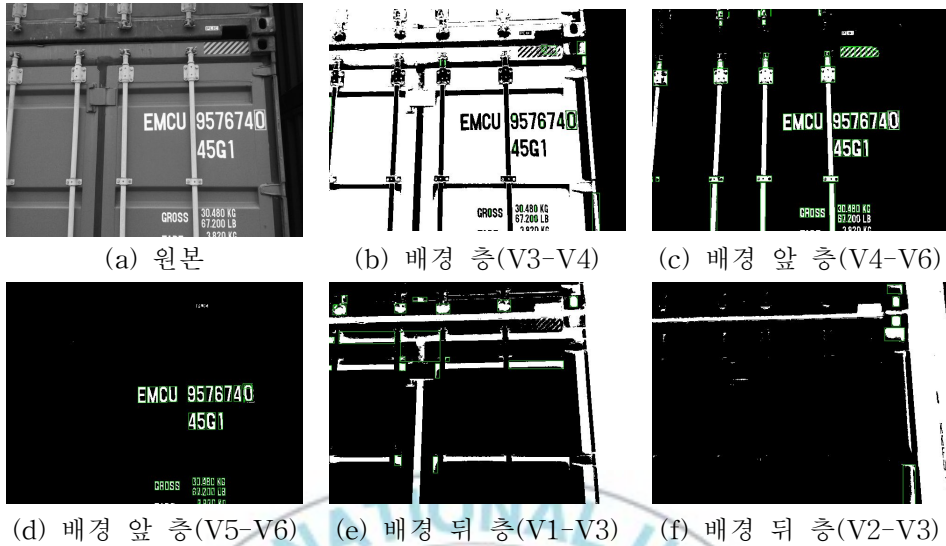
2.2.3 레이블링(Labeling)

레이블링은 이진화영상에서 연결성분을 구분할 때 사용되는 알고리즘으로 인접화소 사이의 영역을 구분할 때 사용한다. 즉 연결성을 판단하여 각각 다른 번호를 부여하여 인접화소의 영역을 구분한다. 레이블링의 8방향의 연결 성분은 픽셀의 대각선 연결을 같은 객체로 인식하는 반면 4방향 연결 성분은 수평이나 수직의 직성 성분만 같은 성분으로 인식한다. 본 실험에서는 4방향의 연결 성분을 사용한다. 그림 18은 레이블링의 예이다.



<그림 18> 레이블링의 예

실제 프로그램에서는 영상에서 레이블 번호를 부여하는 것과 함께, 사각 영역의 좌표정보와 유효픽셀의 개수 그리고 마스크 이미지를 포함한다. 레이블링 알고리즘은 레이블 번호를 부여하는 루프에서 사각 영역의 좌표정보와 유효픽셀 개수를 구해서 계산한다. 그림 18의 생성된 여러 개의 이진화영상을 대상으로 레이블링을 수행한다. 그림 19는 레이블링을 수행한 결과이다.



<그림 19> 레이블링 결과

레이블링의 결과는 영역 정보를 가지고 있는 레이블의 집합으로 리턴 되는데 문자의 가능성이 없는 레이블에 대해 레이블필터(label filter)를 적용하여 선별 작업을 수행한다. 이 때 레이블을 제거하는 기준에는 레이블 개수 제한, 유효픽셀 개수 제한, 가로 최대크기 제한, 세로 최대크기 제한, 가로 최소크기 제한, 세로 최소크기 제한, 가로보다 더 큰 세로 제한, 그리고 가로보다 더 큰 세로 제한 등을 설정할 수 있다.

2.2.4 패넬링(Paneling)

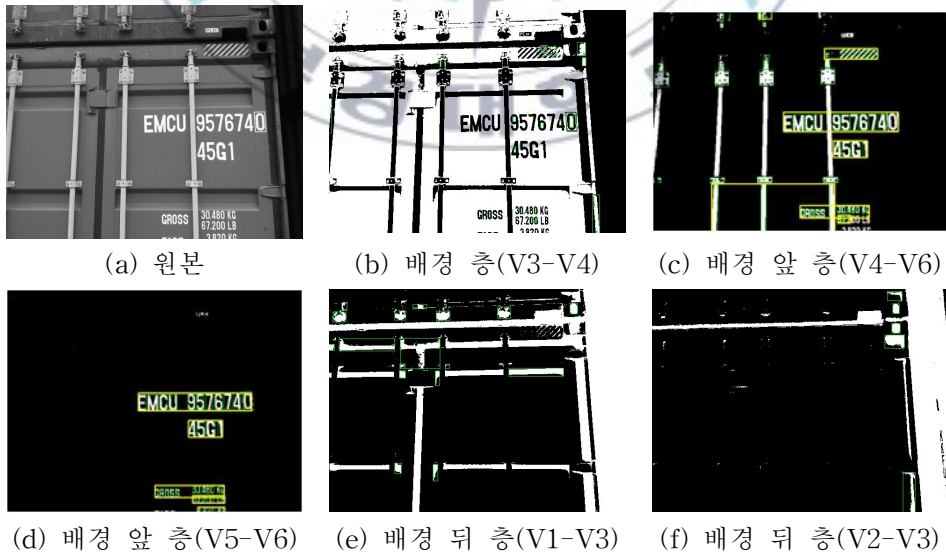
패넬링은 군집상태를 분석하여 문자의 가능성이 있는 개별문자 후보들을 연결하여 패넬형태로 만드는 작업이다. 레이블들의 군집상태에서 수평배열탐색과 수직배열탐색을 수행한다. 군집상태분석은 그림 20의 (a)와 같이 일직선상의 위치여부, 겹침 분석, 최대거리 분석을 통해서, 조건을 만족하는 레이블들을 개별문자 후보로 설정한다. 개별문자 후보로 설정된 레이블들은 그림 20의 (b)와 같이 수평배열탐색으로 레이블 영역을 확장하면서 문자영역을 생성한다. 또한 수

직배열탐색은 세로방향으로 같은 방식으로 수행한다.



<그림 20> 군집상태 분석과 패널링

그림 21은 패널링의 결과이다. 그림 19에서 레이블링을 수행한 영상을 대상으로 패널링을 수행했다. 레이블링에서 시각적으로 군집특성을 보였던 (c)와 (d)에서 정상적으로 패널을 검출했고 (d)에서 가장 선명한 영상을 확보했다. (b), (e), 그리고 (f)는 패널을 검출하지 못하였다.



<그림 21> 패널링 결과

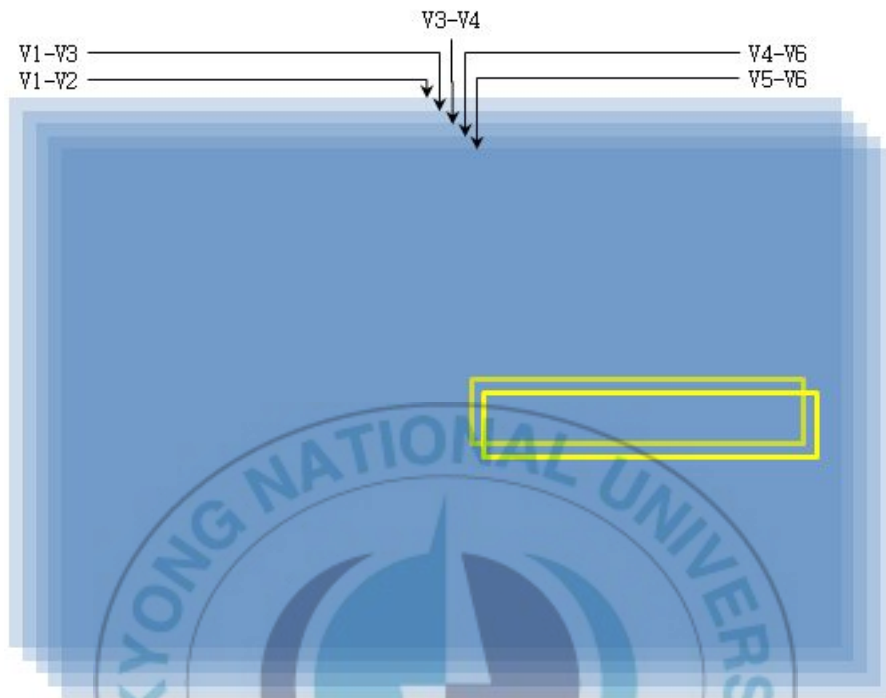
2.2.5 패널병합(Panel Merge)

패널병합은 행간병합, 열간병합 그리고 층간병합으로 구성되며, 문자영역 후보의 수를 줄이는 역할을 한다. 행간병합은 식별자가 여러 행으로 구성된 경우, 같은 층에서 인접한 패널의 병합이며, 패널링의 과정처럼 흡수하며 확장하는 방법이다. 열간병합은 방향만 다르고 같은 방법으로 수행한다. 층간병합은 다른 층에서 겹치는 영역을 하나로 합치는 방법이다. 행간병합을 먼저 수행하고 층간병합을 수행한다. 이는 여러 행으로 분리된 패널들의 임계점은 각각 다를 수 있으며, 같은 영역인데 불구하고 다른 임계치가 적용되면 패널에 따라 다른 결과를 가져올 수 있다. 이를 방지하여 동일한 임계치를 적용시키기 위해 행간병합이 필요하다. 행간병합은 패널들의 시작위치, 높이, 간격 등을 판단하여 2.2.4장의 패널링과 유사한 방법으로 수행된다. 그림 22는 행간병합의 결과이다.



<그림 22> 행간 패널병합

그림 23는 층간병합의 예이다.

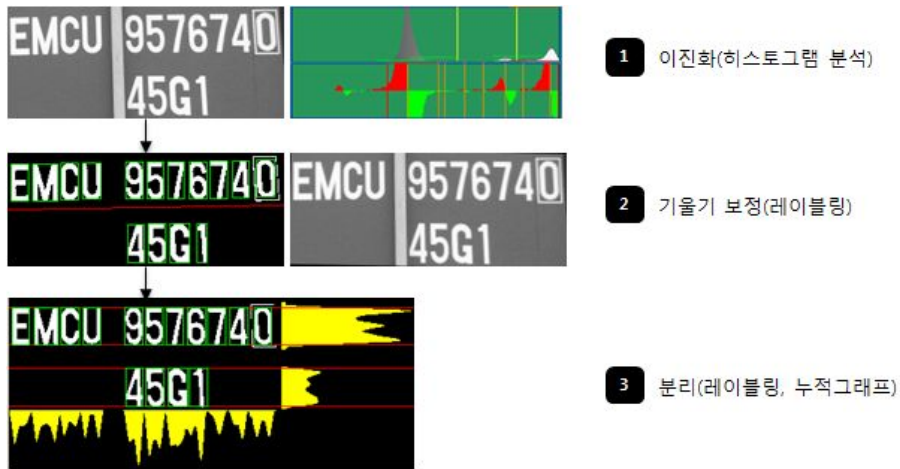


<그림 23> 층간 패널병합

병합패널을 구하는 것이 문자영역의 검출의 마지막 과정이다. 문자영역을 검출하기 위해 다층탐색의 임계치를 사용하였다. 다음 과정에서 문자영역에서 이진화 임계치를 찾을 때 다층탐색의 임계치를 참고하면 더 빠른 처리와 방향성을 쉽게 결정지을 수 있다.

2.3 개별문자 분리

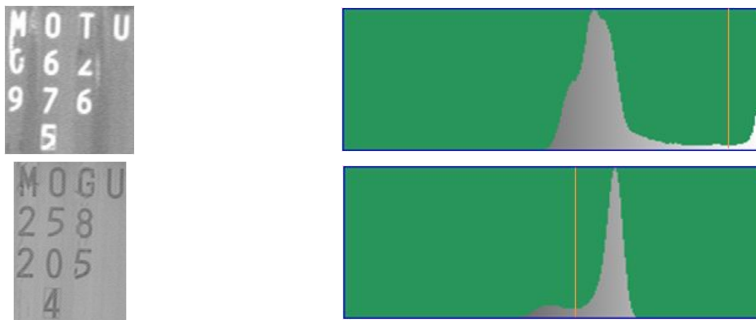
개별문자 분리는 문자영역에서 잡음을 제거하고 배경과 문자를 분리하여 신경망과 같은 분류기의 입력벡터로 만드는 과정이다. 개별문자로 분리하는 과정은 문자영역에서 이진화 임계치를 찾고, 레이블링을 사용하여 중심관계를 분석하고 기울기를 보정한다. 그리고 기울기가 보정된 상태에서 레이블링과 누적그래프의 병행분석으로 잡음제거와 개별문자 분리에 사용했다.



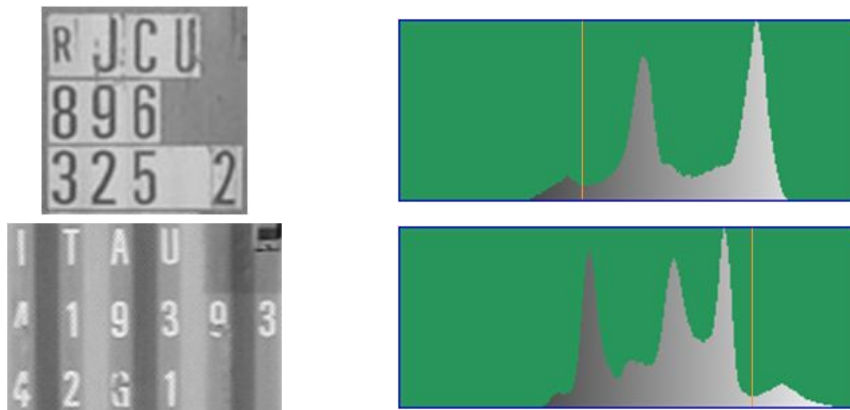
<그림 24> 개별문자 분리 과정

2.3.1 문자영역의 이진화

흑백영상에서 문자영역을 찾기 위해 사용한 이진화의 임계치는 대략적으로 여러 개의 후보를 탐색하였기 때문에 문자영역에서 임계치로 사용될 수 없지만 히스토그램이 양봉일 때 봉의 크기가 비슷해서 어느 방향이 문자인지 판단할 수 없는 경우에는 모호하므로 유용하게 사용된다. 문자영역의 히스토그램은 일반적으로는 양봉형태로 나타나며 단봉이나 다봉형태로도 나타난다. 그림 25는 양봉형태로 임계치를 쉽게 찾을 수 있다. 그림 26은 다봉이지만 문자영역은 가장 밝거나 어두운 봉이기 때문에 임계치를 쉽게 찾을 수 있다.



<그림 25> 양봉형태



<그림 26> 다봉형태



(a) 단봉형태

(b) 임계치 찾기

<그림 27> 단봉형태와 임계치 찾기

그림 27은 단봉형태의 예이다. 문자영역에서 배경과 문자의 명암차이가 두드러지지 않을 때 발생하는 히스토그램의 형태이다. 단봉은 벨리가 없기 때문에 밝기의 시작점과 피크점을 수평이 되도록 회전 시키고, 이동하면서 가장 깊은 지점을 찾아서 임계치로 정할 수 있다. 그림 28은 단봉에서 임계치를 찾는 방법을 나타낸다.

2.3.2 문자영역의 기울기 보정

일반적인 상황에서 컨테이너는 수평을 유지하고 있지만 지면의 상태가 고르지 못하거나, 차량에서 트레일러 구조의 변형, 타이어의 공기압, 카메라가 설치된 구조물의 변형에 따라 수평이 아닌 상황이 발생한다. 그리고 야드트레일러(YT)에 적재되고 정차된 상태에서는 앞쪽이 3°높은 상태로 영상을 획득하게 된다. 이런 이유로 모든 영상에서 회전이 필요하다. 연구 초기 단계에는 컨테이너의 에지(edge)를 찾고, 직선검출알고리즘을 이용하여 회전을 시도하였으나, 영상의 상태에 따라 전혀 검출되지 않는 상황이 있었기 때문에 사용하지 못하였다. 본 논문에서 제시하는 방법은 레이블링의 결과인 레이블들의 집합에서 인접한 레이블과 중심위치의 관계를 분석하여 각각의 기울기를 구하고 그 기울기의 평균으로 하는 전체 기울기를 구해서 회전에 사용하였다. 그림 29는 레이블링을 이용하여 기울기를 보정한 예이다.



<그림 28> 레이블링을 이용한 기울기 보정 알고리즘의 결과

2.3.3 분리(Segmentation)

분리는 문자영역의 이진화 영상에서 개별문자로 떼어내는 과정이다. 영상의 상태에 따라서 다양한 형태를 보이며, 레이블링과 각축의 누적그래프 알고리즘을 적용하여 분리한다. 일반적으로 레이블링만으로 좋은 결과를 보이지만, 다음과 같은 경우는 누적그래프로 분석되어야 한다.

- ① 한 문자가 여러 조각으로 분리
- ② 한 문자 일부분의 손상
- ③ 다른 문자와 붙어있는 경우
- ④ 노이즈 제거

그림 30은 한 문자가 여러 조각으로 분리된 경우이다. 레이블링으로 개별문자 전체를 구하지 못했으며, 누적그래프 분석이 필요하다.



<그림 29> 한 문자가 여러 조각으로 분리된 상태에서의 처리

그림 31은 2행의 마지막 문자가 손상되었다. 이 상태로 레이블링의 결과를 정규화하고 분류기에 넣으면 전혀 다른 결과를 출력할 수도 있다. 누적그래프 분석을 통해 원래의 사각형 영역을 복구하고, 손상되지 않았을 때의 상태로 위치시켜야 한다.



<그림 30> 손상된 상태에서의 처리

그림 32는 일부 식별자가 붙어있는 상태이다. 1차로 누적그래프 분석으로 분리하고 문자수가 모자라거나 기준보다 큰 레이블이 남아있으면 2차로 가로사이즈의 최빈값을 활용하여 통계적인 방법으로 분리한다.



<그림 31> 문자가 붙어있는 상태에서의 처리

그림 33은 잡음으로 인해 레이블링으로 문자를 찾지 못하였다. 누적그래프 분석을 사용하여 잡음을 효과적으로 제거하고, 개별문자의 정확한 범위를 확보한다.



<그림 32> 누적그래프 분석을 잡음 제거에 활용

3. 컨테이너 식별자 영상 인식 및 시스템

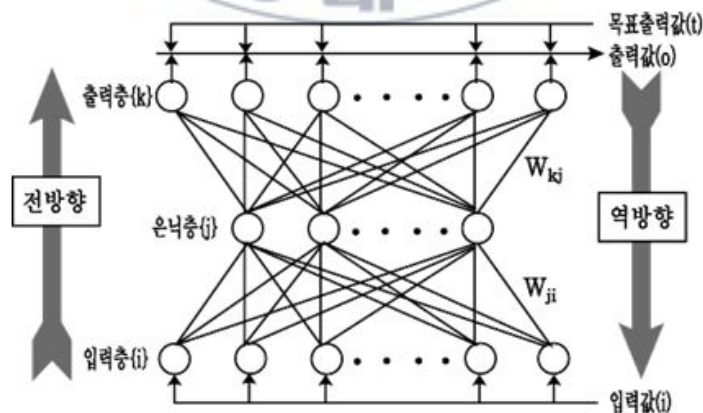
3.1 개별 문자의 인식을 위한 신경망 설계

문자인식은 분리된 개별문자를 분류 인식하는 과정으로 인공신경망(Artificial Neural Network)의 오류역전파알고리즘(Error Back Propagation Algorithm)을 사용했다. 분류기의 입력은 일정한 길이의 값을 요구하므로 개별문자로 분리된 영상은 정규화 과정이 필요하다. 컨테이너 번호는 영문과 숫자로 조합되어있으므로 빠른 학습과 좀 더 명확한 분류를 위해 영문과 숫자의 분류기를 분리하였다. [14]

3.1.1 다층 퍼셉트론

다층 퍼셉트론은 입력층과 출력층 사이에 하나 이상의 중간층이 존재하는 신경망으로 그림 34에 나타낸 것과 같은 계층구조를 갖는다. 이 때 입력층과 출력층 사이의 중간층을 은닉층이라 부른다.

네트워크는 입력층, 출력층, 은닉층 방향으로 연결되어 있으며, 각 층내의 연결과 출력층에서 입력층으로의 직접적인 연결은 존재하지 않는 전방향의 네트워크이다.

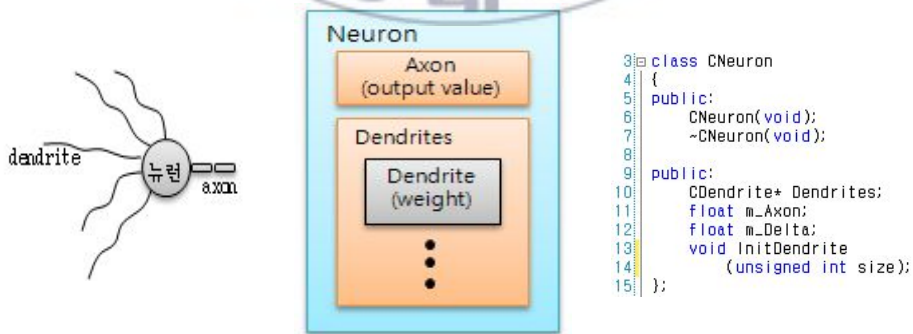


<그림 33> 다층 퍼셉트론의 구조[11]

다층 퍼셉트론은 단층 퍼셉트론과 유사한 구조를 가지고 있지만 중간층과 각 유닛의 입출력 특성을 비선형으로 함으로써 네트워크의 능력을 향상시켜 단층 퍼셉트론의 여러 가지 단점들을 극복했다. 다층 퍼셉트론은 층의 개수가 증가할 수록 퍼셉트론이 형성하는 결정 구역의 특성은 더욱 고급화된다. 즉 단층일 경우 패턴공간을 두 구역으로 나누어주고, 2층인 경우 볼록한(convex) 개구역 또는 오목한 폐구역을 형성하며, 3층인 경우에는 이론상 어떠한 형태의 구역도 형성할 수 있다.

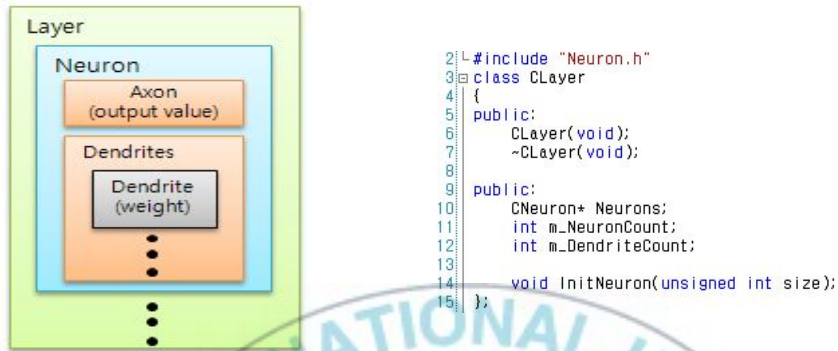
3.1.2 신경망 설계

생물학적 신경망과 다층 퍼셉트론 이론을 바탕으로 하고 객체 지향적 설계방법과 로컬 데이터베이스를 활용하여 신경망을 설계하였다. 뉴런은 다수의 수상돌기에서 정보를 전달받고 활성화여부를 결정하고 활성화 되면 축색돌기를 통하여 다른 뉴런에 정보를 전달한다. 이에 기반 하여 뉴런 클래스의 수상돌기는 입력의 가변을 수용하기 위해 포인터로 선언되고 결과의 보관을 위해 축색돌기를 멤버로 가진다. 뉴런의 설계는 그림 35와 같다.



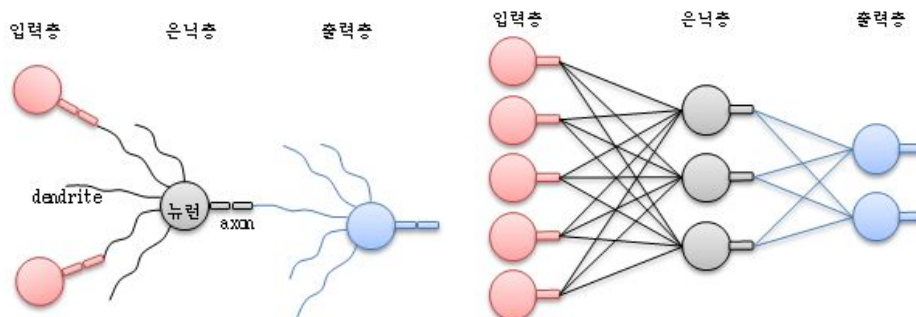
<그림 34> 뉴런의 설계

다층 퍼셉트론은 1개 이상의 은닉층이 존재하며, 본 논문에서는 1개의 은닉층을 사용하였다. 각 층은 여러 개의 뉴런을 가지고 있으며 그림 36은 층을 표현하는 설계이다.



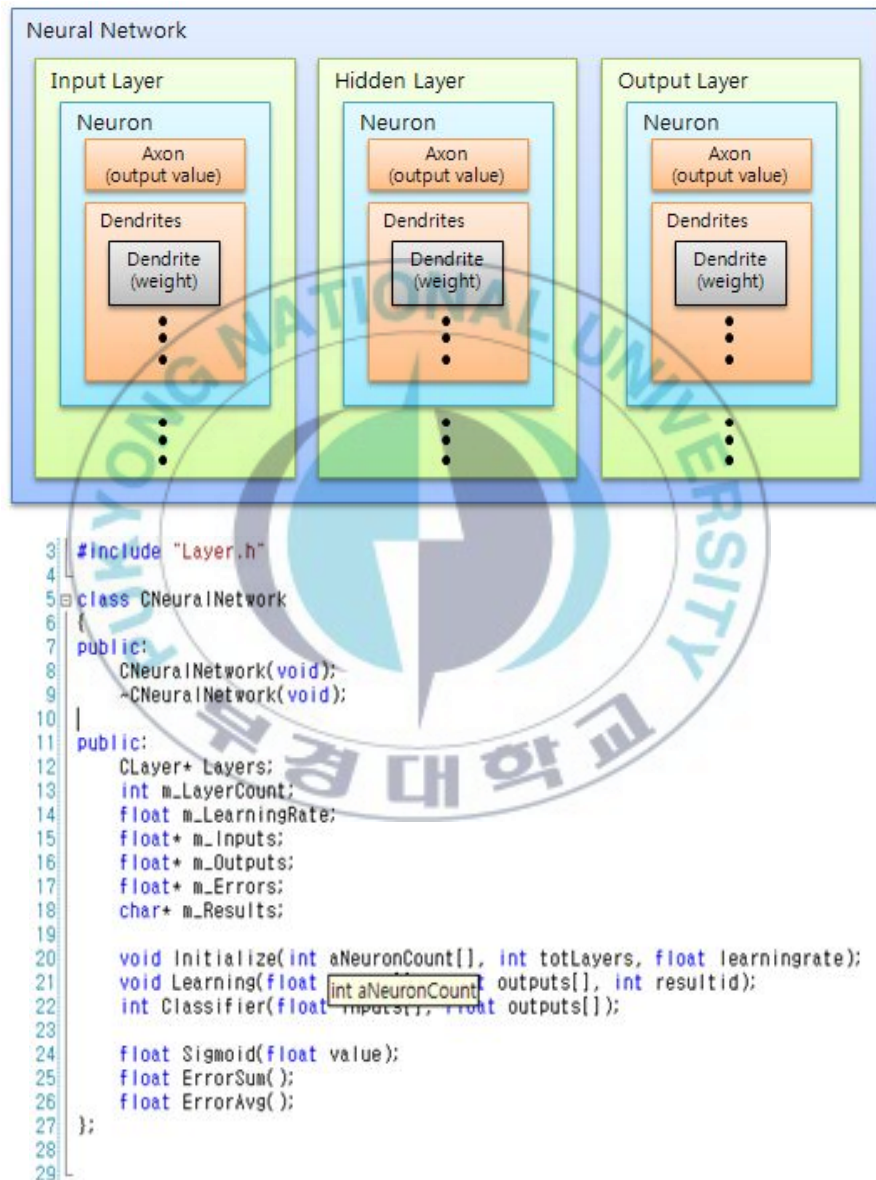
<그림 35> 층을 표현하는 설계

본 연구에서는 입력층, 은닉층, 출력층으로 구성하였다. 입력층의 입력벡터는 수상돌기를 거치지 않고 바로 입력층의 출력(axon)으로 전달되므로 입력층의 수상돌기(dendrite)는 사용되지 않는다. 그림 37은 각층에서 수상돌기와 축색돌기의 소유를 표시한다.



<그림 36> 각층에서 수상돌기와 축색돌기의 구성

신경망클래스는 각 층을 관리하기 위해 가변을 수용할 수 있는 타입으로 선언되어있다. 그리고 학습과 분류를 위한 입력벡터와 출력벡터가 포함되어 있는 등 전반적 관리기능을 포함하고 있다. 신경망클래스의 구조는 그림 38과 같다.



<그림 37> 신경망 클래스

3.1.3. 신경망 데이터 설계

신경망은 학습정보, 입출력패턴, 뉴런의 가중치 등 상당량의 데이터가 발생한다. 파일로 관리할 수 있지만 로컬 데이터베이스에 설계기법을 적용하여 직관적으로 쉽게 알아볼 수 있고 확장이 편리하며 프로그램을 작성할 때 코드를 간결하게 한다.

LEARNING_INFO(정보)	LEARNING_DATA(패턴)	LEARNING_WEIGHT(가중치)
learncode : 학습코드	learncode : 학습코드	learncode : 학습코드
wid : 가로크기	inputdata : 입력패턴	layer_no : 레이어 번호
hei : 세로크기	outputdata : 출력패턴	neuron_no : 뉴런 번호
inp : 입력뉴런 수	resultid : 결과ID	weight : 가중치문자열
hid : 은닉뉴런 수		
out : 출력뉴런 수		
err_avg : 에러평균		
err_max : 에러맥스		
start : 시작시간		
finish : 종료시간		
loopcount : 루프 수		

LEARNING_RESULT(결과)
learncode : 학습코드
resultid : 결과ID
resultchar : 결과문자

<그림 38> 신경망 데이터 설계

그림 39는 신경망의 데이터 설계이다. 빨간색은 PK(primary key)를 의미하며 정보테이블의 학습코드는 패턴과, 가중치, 결과 테이블에서 참조한다. 학습정보 테이블은 여러 개의 학습정보를 관리 할 수 있다. 예를 들어 영문과 숫자를 분리하여 하나의 학습코드로 생성할 수 있고, 영문과 숫자를 분리하여 두 개의 학습코드로 관리할 수 있다. 학습정보 테이블은 학습에 관한 기본적인 정보를 담

고 있다. 정규화 사이즈나 뉴런 수, 학습 종료의 기준이 되는 에러평균이나 에러맥스 그리고 학습에 소요된 시간이 얼마나 걸렸는지 조회한다. 패턴테이블은 입력벡터와 출력벡터를 관리한다. 학습코드와 입력패턴, 출력패턴이 PK로 잡혀 있기 때문에 동일한 벡터가 입력되면 데이터베이스 시스템에서 오류를 알려준다. 그리고 입력패턴은 같은데 출력패턴이 다를 경우도 새로운 패턴을 입력하는 시점에서 간단한 쿼리로 조회가 가능하다. 패턴테이블의 결과ID는 출력벡터가 어떤 문자인지 ID형태로 존재하며 해당 ID를 이용하여 결과테이블에서 실제의 문자를 조회한다. 가중치테이블은 은닉층과 출력층의 가중치를 문자열 형태로 보관한다. 은닉층 수상돌기 가중치의 수는 입력층의 뉴런수와 같고 출력층 수상돌기 가중치는 은닉층의 뉴런수와 같다. 입력층은 수상돌기를 사용하지 않으므로 가중치가 존재하지 않는다.

표 1은 가중치 테이블의 예이다. NUM01학습코드의 1번레이어 87번 뉴런의 가중치 정보를 담고 있다. 결과테이블은 패턴테이블의 결과 ID와 매핑된 텍스트 정보를 제공한다.

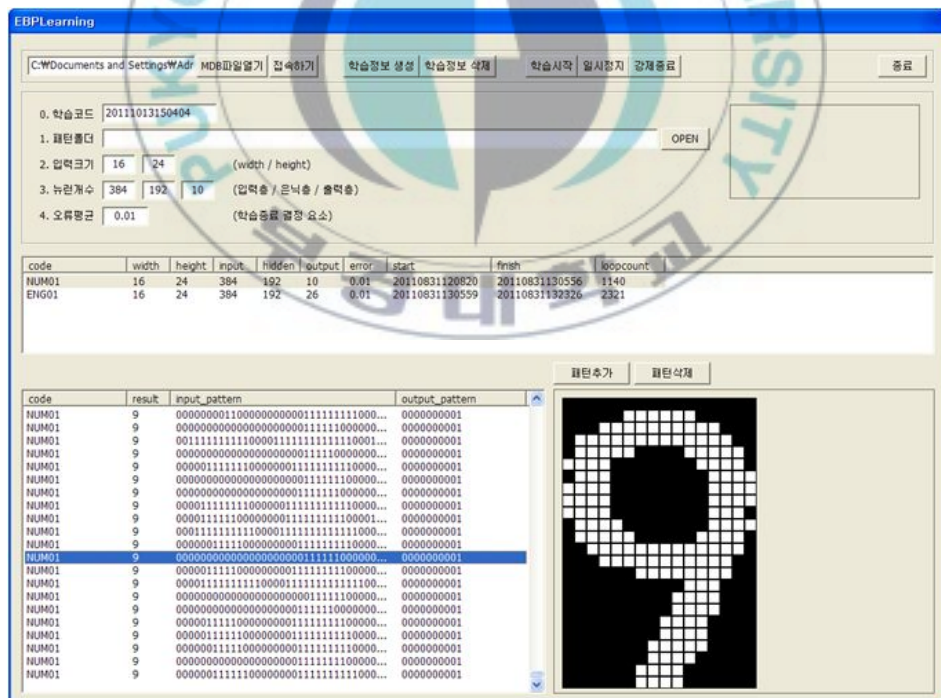
<표 1> 가중치테이블의 예

learncode	layer	neuron	weight
NUM01	1	87	-0.4172;-0.1002;0.3046;-0.2564;-0.1126;0.3875;0.4009;0.1992;0.4151;-0.0478;0.1646;0.4067;-0.4618;0.3383;0.0343;-0.3745;-0.2142;-0.5343;-0.1244;0.1578;0.0590;0.1515;0.3192;0.1498;0.1188;0.4061;-0.3934;-0.2571;0.3637;-0.1690;-0.0694;0.4840;-0.4490;0.2230;-0.2716;0.1174;0.0343;0.4101;-0.3576;-0.1436;0.3158;-0.1907;-0.0679

표 2는 결과테이블의 데이터가 입력된 예이다.

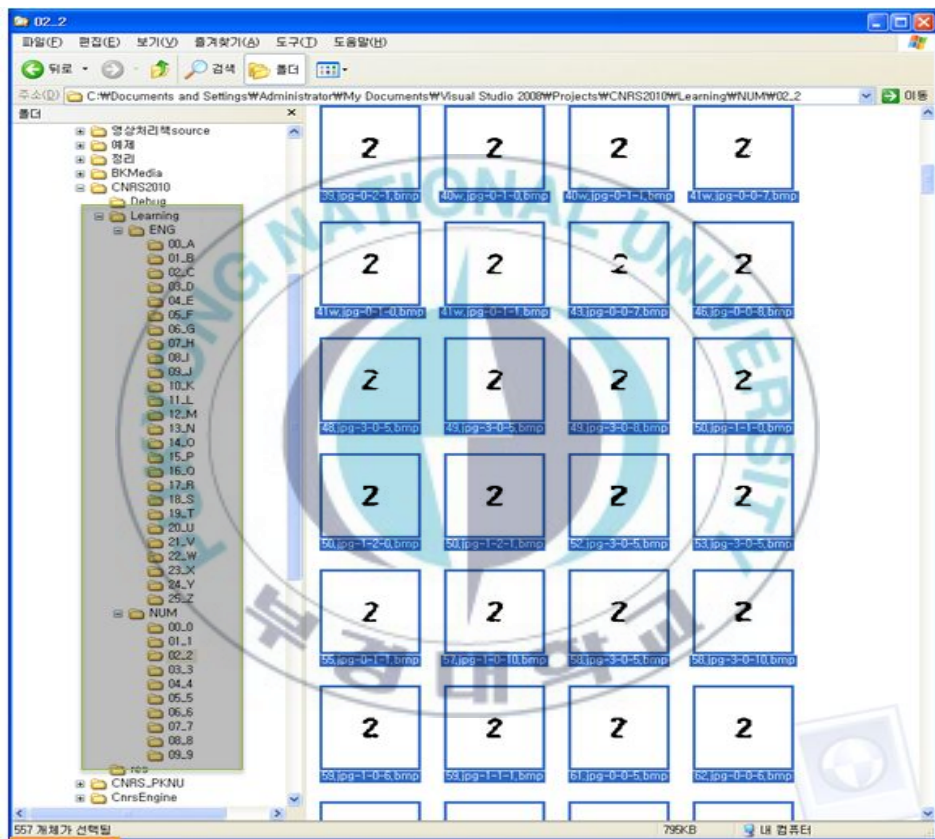
<표 2> 결과테이블의 예

learncode	resultid	resultchar
ENG01	0	A
ENG01	1	B
ENG01	2	C
ENG01	3	D
ENG01	4	E
ENG01	5	F



<그림 39> 학습 프로그램

그림 40은 학습 프로그램은 신경망의 클래스와 데이터 설계를 기반으로 동작하는 프로그램이다. 학습코드를 정하고 패턴폴더에서 입력이미지 폴더를 선택하고 입력크기와 뉴런 수 그리고 오류평균을 입력하여 학습정보를 생성시킨다. 여러 개의 학습정보를 생성시킬 수 있고 학습코드에서 패턴을 추가하거나 삭제할 수 있다.



<그림 40> 패턴 폴더

그림 41은 학습 프로그램에서 관리하는 패턴 폴더이다. 프로그램에서 패턴 폴더를 입력하면 "ENG"와 같은 상위폴더는 자동으로 학습코드로 지정이 되고 하위폴더의 "ID_TEXT" 구조는 자동으로 결과테이블을 만든다.

3.1.4 정규화

정규화는 그림 42와 같이 신경망 처리에서 입력벡터와 동일한 사이즈로 바뀌 주는 작업이다. 1이나 I와 과 같이 가로의 폭이 좁은 문자를 정규화 할 경우 정규화 영역이 유효픽셀로 가득 차는 경우가 생길수도 있으므로 가로의 폭이 좁은 문자는 임의로 배경을 넣어 정규화 한다.

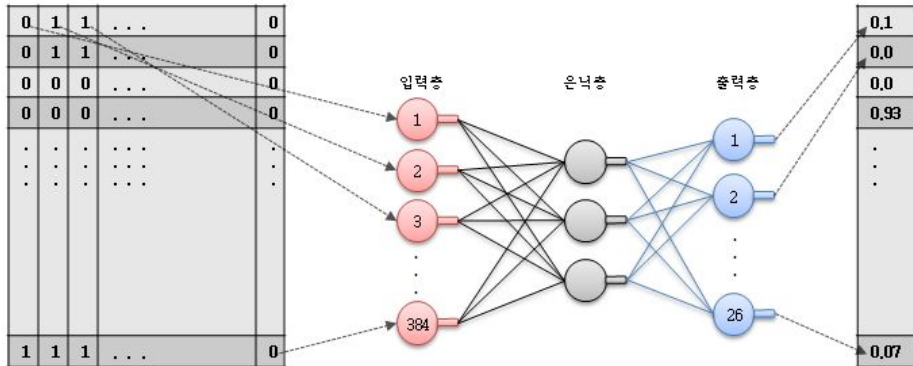


<그림 41> 정규화의 예

3.1.5 문자인식

문자영역에서 분리과정과 정규화를 거친 개별문자는 신경망에서 지정한 일관된 사이즈이며 신경망의 입력패턴으로 사용할 수 있다. 인식 과정은 신경망이 사전에 학습이 된 상태를 전제로 한다. 학습이 된 상태는 학습코드가 존재하고 입출력 패턴에 의해 가중치가 결정된 상태이다.

컨테이너 번호는 영문과 숫자로 구성되어 있다. 영문과 숫자를 하나로 합쳐서 학습을 해도 되지만 숫자0과 영문O, 숫자 1과 영문 I등 몇몇 겹치는 비슷한 문자, 패턴이 추가, 삭제되었을 때 학습시간 등으로 인해 숫자신경망과 문자신경망을 분리해서 사용한다. 분리과정에서 이미 배열 상태나 문자의 개수를 파악하기 때문에 어디까지가 문자이고 숫자인지는 대부분 파악이 되나 문자의 개수가 11보다 작거나 클 경우는 문자와 숫자의 위치가 명확하지 않기 때문에 숫자와 문자 두 개의 신경망에 모두를 사용하고 높은 인식률을 참고해서 결정한다. 실험에서는 16 x 24 사이즈를 정규화에 사용하였으므로 입력패턴은 384개이며 숫자의 출력패턴은 10개이고 영문자의 출력패턴은 26개이다.



<그림 42> 신경망의 입력패턴과 출력패턴

그림 43은 신경망의 입력패턴과 출력패턴의 예이다. 입력층에는 정규화 된 2 차원 이미지를 1차원으로 변경하여 입력패턴으로 지정하고 은닉층을 거쳐 출력층의 결과는 0이상 1이하의 값으로 출력이 되는데 일반적으로 가장 큰 값을 출력의 결과로 결정한다. 영문자는 출력패턴은 26개이다. 0번째를 시작으로 3번째가 0.93이므로 가장 큰 값이므로 출력값이 되고, 결과ID는 3이다. 결과테이블에서 해당 학습코드의 결과ID가 3인 resultchar를 조회하면 인식결과는 D가 될 것이다.

4. 실험 결과 및 고찰

4.1 실험 환경

실험한 시스템 환경은 인텔 펜티엄 듀얼코어 E7300, Windows XP, 2GB RAM, Visual Studio 2008에서 MFC Visual C++로 프로그래밍하였다.

그림 44는 실험에 사용한 분석 프로그램으로 한 장의 영상을 분석하는 과정을 예시로 표시하였다.

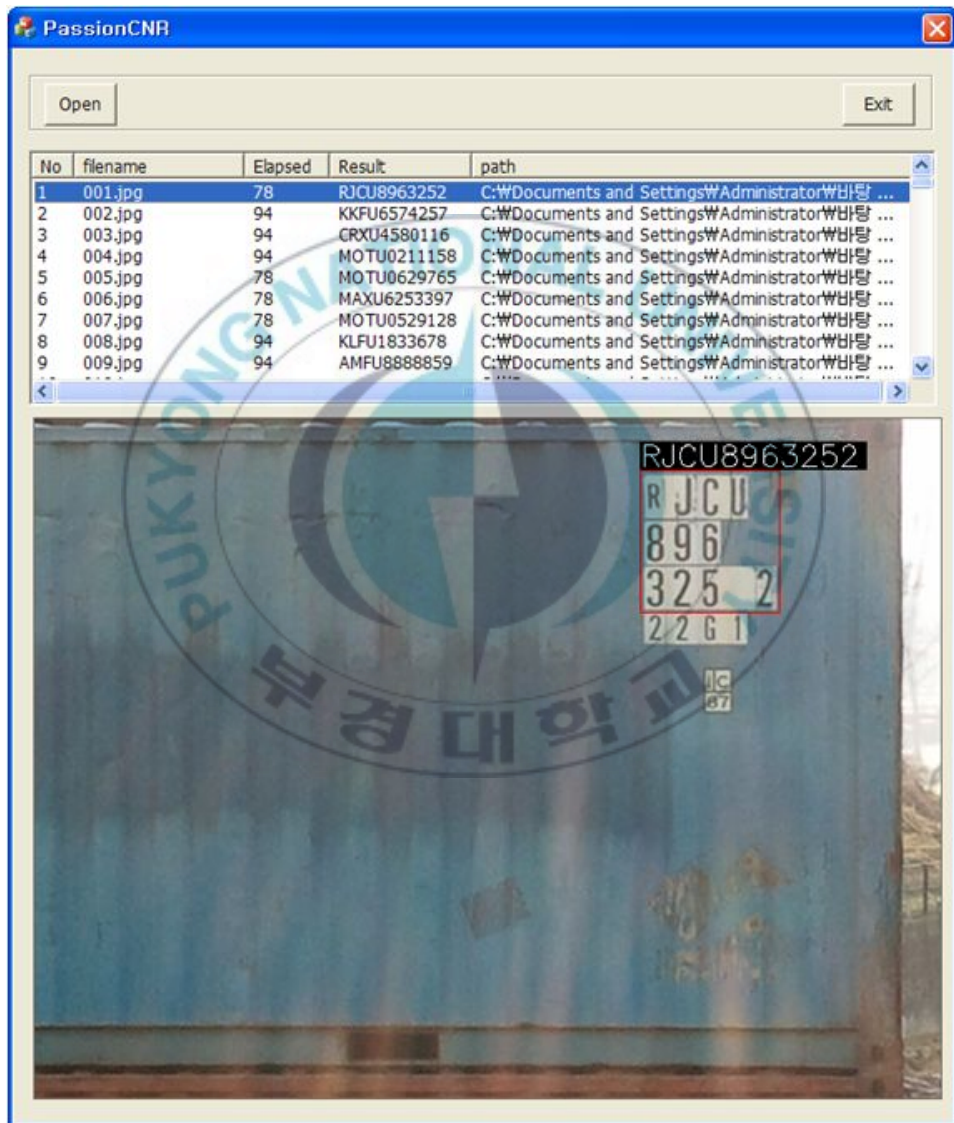


<그림 43> 실험에 사용한 분석 프로그램

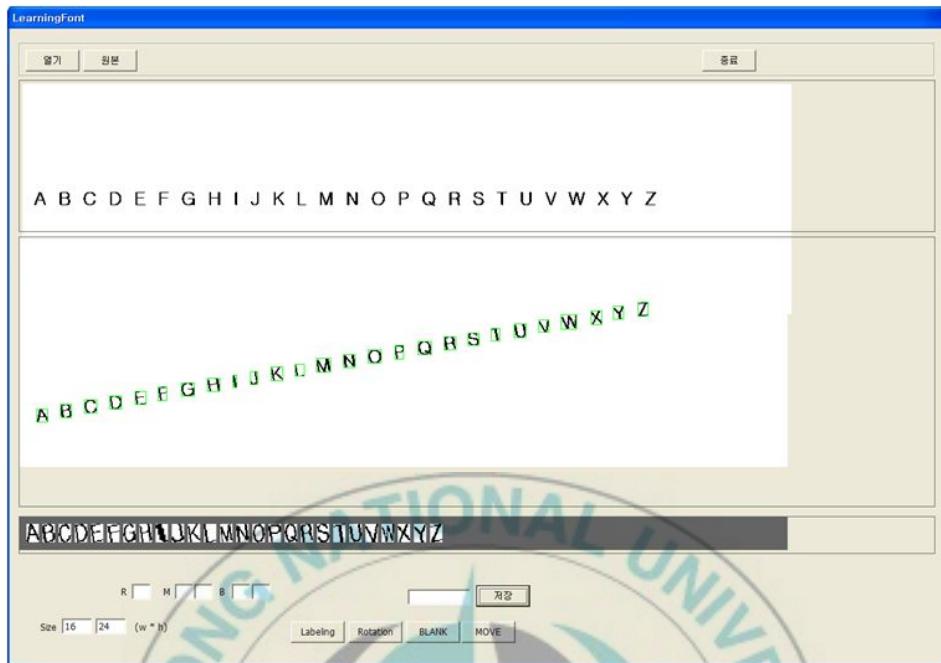
그림 45는 실험에 사용한 결과 프로그램이다. 여러 영상의 결과를 빠르게 보기 위해 개발하였다. 영역검출의 결과를 표시하고, 인식 결과와 수행시간을 보여준다.

그림 46은 실험에서 인식률을 높이기 위해 다양한 패턴을 만들어 학습시킬 수 있는 학습폰트 작성기이다. 좌·우 회전하면서 사각형을 조금 벗어난 경우와

공백이 있는 경우 등 문자 1개당 187개의 다른 패턴을 제공한다. 테스트 환경에서 많은 영상을 구할 수 있다면 해당 영상에서 추출한 문자로 학습하는 것이 가장 좋겠지만, 영상 샘플이 부족한 경우 학습폰트 작성기는 많은 도움이 된다. 그림 47은 학습폰트 작성기 결과의 일부이다.



<그림 44> 실험에 사용한 결과 프로그램



<그림 45> 학습폰트 작성기



<그림 46> 학습폰트 작성기 결과의 일부

4.2 신경망의 학습 정보

신경망의 학습은 영문과 숫자를 분리해서 영문신경망과 숫자신경망으로 구분하였다. 크기는 16 x 24로 정규화했고, 입력층은 384개의 뉴런, 은닉층은 192개의 뉴런을 사용한다. 출력층은 영문 대문자 26개의 뉴런으로 구성되고 숫자는 10개의 뉴런으로 구성된다. 학습에 사용된 개별문자 패턴의 개수는 영문은 300

여개, 숫자는 500개 이상 패턴으로 학습하였다. 학습종료 결정요소는 둘 다 가장 큰 오차가 1% 미만으로 설정하였다. 그리고 영문코드의 학습시간은 약 2시간 소요되었고 2321회의 반복학습을 수행하였고, 숫자코드의 학습시간은 약 1시간에 걸쳐 1140회의 반복학습을 수행하였다.

code	width	height	input	hidden	output	error	start	finish	loopcount
NUM01	16	24	384	192	10	0.01	20110831120820	20110831130556	1140
ENG01	16	24	384	192	26	0.01	20110831130559	20110831132326	2321

<그림 47> 인공신경망의 학습 정보

4.3 컨테이너 식별자 인식 실험

컨테이너 식별자 인식의 실험에는 (a) 해외인식기에서 실패한 측면 영상 105장, (b) 직접 촬영한 측면 영상 38장, (c) 직접 촬영한 앞면 8장, (d) 직접 촬영한 뒷면 11장 등 162개의 샘플영상을 사용하였다. 문자영역 검출 및 개별문자 분리, 인식 부분으로 나누어서 중간 처리과정의 결과는 표3과 같다.

전체 영상에서 육안인식 가능 기준은 컨테이너 번호 11문자 중에서 1개의 문자라도 손실 또는 망실되면 육안인식 불가로 분류하였다. 표 1의 (a)는 해외 상용제품에서는 실패한 영상을 대상으로 실험한 결과로 인식률은 80%이고, 육안으로 인식 가능한 영상의 인식률은 91.3%를 보였다. (b), (c), 그리고 (d)는 직접 촬영한 상태가 좋은 경우로 100%의 인식률을 보였다.

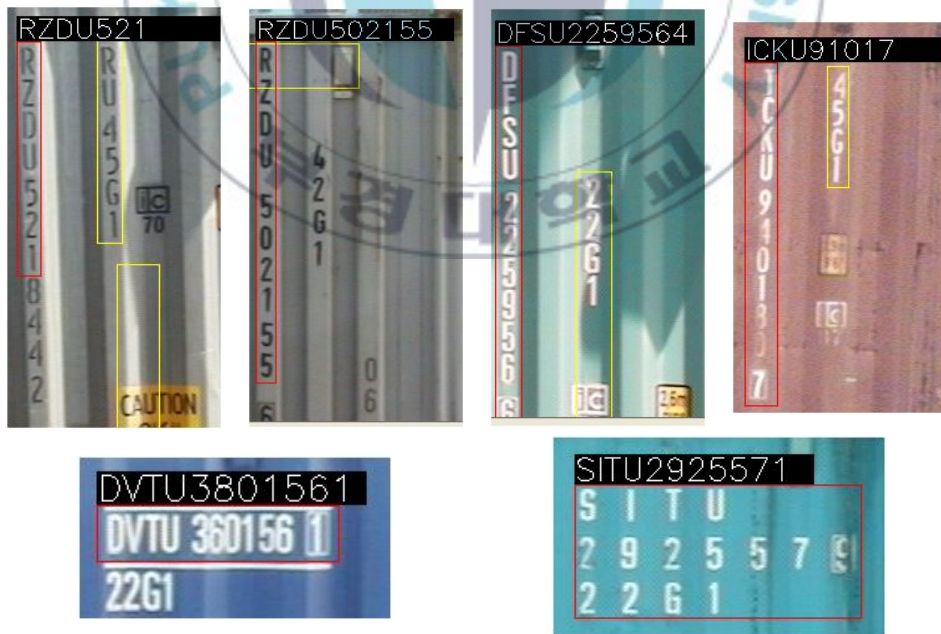
그림 49는 육안인식이 가능하나 인식에 실패한 영상으로 영상 획득 시 문자의 일부분이 잘리거나 손상된 경우가 주요 원인이다. 그리고 문자와 배경의 명암대비가 적어서 z축 누적그래프에서 특징이 드러나지 않거나, 개별문자에서 명암차

이로 인해 이진화에서 일부분이 누락된 상태로 다른 문자로 인식되거나 인식하지 못한 경우이다.

<표 3> 실험 결과 인식률

영상구분	육안인식 가능(%)	영역검출 (%)	분리(%)	인식(%)	육안인식 인식(%)
(a) 영상	87.61	99.04	98.09	80	91.3
(b) 영상	100	100	100	100	100
(c) 영상	100	100	100	100	100
(d) 영상	100	100	100	100	100
전체	91.97	99.38	98.76	87.03	94.63

그림 50은 육안인식도 불가능한 영상으로 컨테이너는 외부 환경, 주야간, 외부 오염 등과 같은 요소로 인하여 손실된 경우이다.



<그림 48> 육안인식이 가능하나 실패한 경우



<그림 49> 육안인식이 불가능한 경우

5. 결론

항만을 효율적으로 운영하기 위해 자동화나 무인화에 관련된 기술 중에서 컨테이너 식별자 영상 인식은 게이트나 야드, 본선에서 모두 적용 할 수 있다. 본 논문에서는 컨테이너 식별자의 영상 인식을 연구하였다. 특히 상용인식 제품에 서 실패한 영상을 중점적으로 연구하였다. 전체 처리 과정은 문자영역검출과 개별문자분리 그리고 문자인식으로 구성하였다. 문자영역검출에서는 이진화의 임계치를 찾기 위해 흑백영상을 3차원으로 구성해서 분석하는 다층탐색기법을 적용하여 이진화 후보영상 생성하고, 레이블링과 패넬링, 그리고 패넬병합 과정을 거쳐 문자영역을 검출하였다. 개별문자분리에서는 문자영역에서 레이블링을 수행하고 레이블들의 중심관계를 분석하여 영상의 기울어짐을 보정하며, 레이블과 각 축의 누적그래프의 병행분석으로 개별문자 분리와 잡음제거에 적용하였다. 문자인식은 인공신경망의 오류역전파알고리즘을 사용하였다. 새로운 패턴이 추가되었을 때 빠른 학습과 좀 더 명확한 분류를 위해 영문과 숫자로 분리하여 인식시켰다.

실험결과는 기존 상용엔진에서 실패한 영상에서 80%의 인식률을 보였다. 이 중 육안으로 식별이 가능한 영상에 대해서는 91.3%의 인식률을 보였다. 상용엔진에서 실패한 영상임을 감안하면 우수한 인식률이다. 그리고 육안으로 확인 가능하고 선명한 영상은 100% 인식으로 우수한 인식률을 보였다. 향후 다양한 환경에서의 영상들을 대상으로 인식률을 향상시키는 연구가 요구된다.

참고문헌

- [1] 기술표준원, “KS 국제 화물 컨테이너의 기호, 식별 및 표시 방법”, KS T3201, 한국표준협회, 2007.
- [2] 최창운, “컨테이너 번호 추출을 위한 영상 처리 시뮬레이터 설계 및 구현”, 한국시뮬레이션학회논문지, 제9권 제3호, pp.53-64, 2000.
- [3] 김광백, “형태학적 특성과 FCM 기반 퍼지 RBF 네트워크를 이용한 컨테이너 식별자 인식”, 한국해양정보통신학회논문지, 제11권 제6호, pp.1162-1169, 2007.
- [4] 구경모, “항만 영상정보시스템 구축을 위한 컨테이너 식별자 인식”, 한국해양정보통신학회 2007년도 춘계종합학술대회, 2007.
- [5] 최광학, “컨테이너 터미널 자동화 게이트시스템 기술적용 연구”, 인하대학교 석사학위 논문, 2008.
- [6] 조영수, “RFID기반 컨테이너 터미널 게이트 반출·입 자동화 설계”, 숭실대학교 석사학위 논문, 2011.
- [7] 신중조, “RFID/OCR기반의 자동화 게이트시스템 기술 개발에 관한 연구”, 동아대학교 석사학위 논문, 2007.
- [8] 최상희, 심기섭, 김우선, 하태영, “국내 컨테이너 항만기술개발 로드맵 수립 연구, 항만물류 및 장비기술을 중심으로”, 한국 해양수산개발원, 2007.
- [9] 조재현, “자동차번호판 이진화를 위한 개선된 퍼지 이진화 방법”, 한국전자통신학회논문지, 제6권 제2호, pp.231-236, 2011.
- [10] 김민기, “자동차번호판 이진화 방법에 대한 성능비교”, 한국콘텐츠학회논문지, 제9권 제8호, pp9-17, 2009.
- [11] 서석태, 권순학, “그레이 레벨의 공간적 분포에 기반한 영상 이진화 기법” 한국퍼지및지능시스템학회 춘계학술대회 학술발표 논문집, pp.265-268, 2005.

- [12] 서병훈, 김병만, 문창배, 신윤식, "그림자가 있는 차량 번호판의 이진화", 한국산업정보학회논문지, 13권, 제4호, 2008.
- [13] N. Otsu, "A Threshold Selection Method from Gray-level Histograms", IEEE Trans. on SMC, Vol.9, No.1, pp.62-66, 1979.
- [14] 김대수, "신경망 이론과 응용 I", Jinhan M&B, 2005.
- [15] Hearn, Baker, "Computer Graphics with OpenGL", Prentice Hall, 2004.
- [16] 황선규, "영상처리 프로그래밍 by Visual C++", 한빛미디어, 2009.

