

Back-propagation 알고리즘을 이용한 컨테이너 식별자 인식에 관한 연구

이만형^{*}, 황대훈
 경원대학교 전자계산학과

A Study on the Container Identifier Recognition using Back-propagation Algorithm

*ManHyoungh Lee, DaeHoon Hwang

Dept. of Computer Science, Kyungwon Univ.

요 약

오늘날 항만에서의 컨테이너 물류처리 방법은 수작업의 형태가 주류를 이루고 있다. 그러나 과도한 물동량의 증가로 인하여 항만에서는 효율적인 컨테이너 처리를 하지 못하고 있는 가운데 새로운 컨테이너 물류처리 방법의 필요성이 대두되었다.

이에 본 논문에서는 컨테이너의 자동화 처리를 위해서는 식별자의 인식이 가장 큰 문제점을 해결하기 위해 항만에서 컨테이너의 식별자 인식에 신경망 알고리즘의 하나인 Back-propagation을 사용하여 기존의 식별자 인식 방법보다 신속하고 정확한 처리가 가능하도록 구현하였다.

로 인하여, 오늘날 항만에서의 컨테이너 처리를 위한 시스템으로는 영상 처리 컨테이너 식별자 인식 시스템을 적용하는 추세에 있다.

영상 처리 컨테이너 식별자 인식 시스템은 기존의 설비 장비만으로 모든 번호판의 판독이 자동으로 처리될 수 있으며, 신규 컨테이너 증가와 번호판 규격의 변경 시에도 일부 소프트웨어의 변경만으로 기존 장비를 100% 활용할 수 있다는 장점을 가진다.

이에 본 연구에서는 항만에서 컨테이너의 선적 및 야적 등과 같은 물류처리를 자동화하기 위하여 신경망 알고리즘의 하나인 Back-propagation을 식별자 인식에 적용함으로써 보다 정확하고 신속한 식별자 인식이 가능하도록 하였다.

I. 서론

현재 일부 항만에서 컨테이너의 자동 처리를 위해 채택되고 있는 시스템은 크게 바-코드(bar-code) 시스템과 영상 처리를 기반으로 하는 컨테이너 식별자 인식 시스템으로 대별된다.

바-코드 시스템은 컨테이너에 바-코드를 부착하거나 소지하여 처리하는 것으로 훼손이나 분실시에는 컨테이너를 자동으로 처리할 수 없을 뿐 아니라, 차량 번호판이나 컨테이너 ISO 번호가 변경될 때마다 바-코드를 교체해야 하는 문제점 등을 갖고 있다. 따라서 이 시스템은 추가 경비 지출이 계속하여 발생하고 관리비용이 과다하게 지출되는 문제점 등으

II. 컨테이너 식별자

2.1 컨테이너 식별자의 ISO 규격

컨테이너의 식별자 코드는 다음과 같은 다섯 개의 구분된 영역으로 구성되어 있다. 이 중 Owner code는 국제 컨테이너 관리국에 의해 전세계적으로 고유하게 부여된 4개 문자 코드이다.

- > Owner Code : 4자의 영문자(4번째 문자는 특수 코드로 항상 "U", 규정 사항)
- > Serial Number : 6자의 숫자(규정 사항)
- > Check Digit : box 내의 1자의 숫자(규정 사항)
- > Country Code : 2자의 영문자(특수 코드로 선택 사항)
- > Site and Type Code : 4자의 숫자(특수 코드로 선택 사항)

※ 본 연구는 한국과학재단의 산학협동연구 (과제번호 97-2-11-(02-01-1)의 지원에 의한 결과임

2.2 컨테이너 식별자의 유형 및 특징

모든 컨테이너에 고유하게 부여된 식별자는 4개의 영문자와 체크 숫자 1개를 포함한 7개의 숫자 등 모두 11자의 문자로 구성된다.

컨테이너에 기록하는 문자 유형에는 어떤 표준이나 강제적인 규약이 없으므로, 어떤 문자는 이런 폰트를 쓴다고 한 마디로 단언할 수는 없다. 그리고 이러한 식별자들은 위의 규정을 제외하고는 특별히 정형화되어 있지 않기 때문에 인식에 있어 어려운 점이 되고 있다.

또한 문자들은 모두 컨테이너의 외부에 노출되어 있기 때문에, 문자의 유실이나 오물 등과 같은 잡음으로 인하여 문자를 인식하는데 어려움이 따른다. 아울러 컨테이너의 바탕 즉 문자가 쓰여있는 부분이 불균형되어 있는 특징으로 인하여 문자의 왜곡 등과 같은 형태 변형과 명암 등으로 인한 문자의 잡음 개입과 유실 등의 영향을 미칠 수 있다.

III. 본 연구의 특징

3.1 Back-propagation 식별자 인식 방법의 특징

Back-propagation을 이용한 컨테이너 식별자 인식 방법은 다음과 같은 특징을 가지고 있다.

1. 신경망을 이용하는 방법은 다른 방법보다 패턴 인식에서 탁월한 인식능력을 갖고 있다.
2. 에러값을 갖고 있어서 학습시킨 패턴과 달리 입력된 문자값에 노이즈를 갖고 있거나 차이가 있어도 이를 허용범위 내에서 극복할 수 있다.
3. 새로운 문자 패턴이 발생하여도 새롭게 알고리즘의 보안이나 추가 없이 학습을 시켜주면 되므로 유지 보수가 용이하다.
4. 단일인식으로 안되더라도 학습패턴과 유사한 것을 구할 수 있다.
5. 학습시간이 다소 많이 소요되지만 큰 문제는 되지 않으며 실행시에 빠른 결과를 얻을 수 있다.

3.2 Back-propagation의 구성

Back-propagation의 구성은 학습을 위한 패턴 파일과 학습의 결과를 갖고 있는 학습 파일을 가지고 있으며, 숫자의 경우 입력노드 200개, 미들 노드 100, 출력 노드 10개, 알파벳의 경우 두 개의 뉴럴넷으로 구성되어 출력 노드가 각각 13개씩이 된다.

컨테이너 문자는 4개의 알파벳과 7개의 숫자로 구성되어 있으므로 이를 하나의 뉴럴넷으로 묶어서 학습하는 경우와 각자 다른 뉴럴넷을 구성하여 학습시키는 방법이 있다.

(1) 묶어서 학습시킬 경우

이렇게 하게 되면 전처리에서 각자 분할이 잘못되거나 숫자나 알파벳의 구분 없이 인식하므로 이러

한 문제를 고려하지 않아도 된다. 그러나 같이 학습시킬 경우 알파벳과 숫자사이의 유사문자에 영향을 주게 되며 전체적으로 인식률을 낮추게 될 우려가 있다. 또한 알파벳을 숫자로 숫자를 알파벳으로 인식하는 경우가 발생할 수 있다. 학습을 하는데 소요되는 시간도 커지며 알파벳만을 추가적으로 학습시킬 때 전체적으로 모든 문자를 학습시켜야 하는 불합리한 점이 발생한다.

(2) 각자 다른 신경망으로 구성하는 경우

유사한 문자를 배제 할 수 있어 묶어서 하는 경우보다 안정적으로 인식할 수 있으며 인식률도 높다. 그러나 전처리의 오류로 숫자에 해당하는 부분을 알파벳을 인식하기 위한 뉴럴넷으로 넘겨주면 100% 오인식한다. 학습을 시킬 경우 따로 학습을 시키므로 학습시간을 크게 단축시킬 수 있다.

3.3 수행 절차

먼저 분할된 문자를 입력받는다. 입력받은 문자를 10*20으로 정형화시키고 해당 뉴럴넷으로 입력되어 출력값을 구한다. 그리고 출력된 코드값은 3개의 형태 문자로 매칭 시킨다.

① 출력값은 3개의 형태로 구분한다.

- 단일인식 : 만족하는 값이 하나일 경우
 - 중복인식(+) : 만족하는 값이 두 개 이상일 경우
 - 무인식(-) : 만족하는 값이 아무 것도 없을 경우
- ② 인식률은 3개의 형태순이며 중복인식과 무인식의 경우 가장 높은 유사도를 갖는 값을 인식문자로 출력하였다.

다음으로는 후처리과정을 수행하는데 다음 그림과 같다. 다음 그림에서 알파벳 인식을 두 부분으로 나눈 이유는 유사한 문자로 인해 오인식 할 확률을 줄이기 위해서이다.

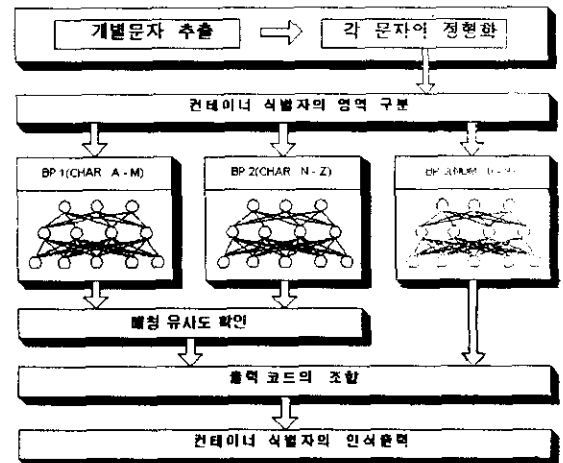


그림 3.1 Back-propagation 수행과정

패턴	Target Value	10진값	패턴	Target Value	10진값
0	0000000001	1	5	0000100000	32
1	0000000010	2	6	0001000000	64
2	0000000100	4	7	0010000000	128
3	0000001000	8	8	0100000000	256
4	0000010000	16	9	1000000000	512

(a) BP1 학습 패턴의 목표값

패턴	Target Value	10진 값	패턴	Target Value	10진 값
A	0000000000001	1	H	0000010000000	128
B	0000000000010	2	I	0000100000000	256
C	00000000000100	4	J	0001000000000	512
D	00000000001000	8	K	0010000000000	1024
E	0000000010000	16	L	0100000000000	2048
F	0000000100000	32	M	1000000000000	4096
G	0000001000000	64			

(b) BP2 학습 패턴의 목표값

패턴	Target Value	10진 값	패턴	Target Value	10진 값
N	00000000000001	1	U	0000010000000	128
O	00000000000010	2	V	0000100000000	256
P	000000000000100	4	W	0001000000000	512
Q	0000000000001000	8	X	0010000000000	1024
R	00000000010000	16	Y	0100000000000	2048
S	00000000100000	32	Z	1000000000000	4096
T	00000001000000	64			

(b) BP3 학습 패턴의 목표값

표 3.1 학습 패턴의 목표값

Target Value는 보통의 경우에는 4개로도 충분히 각 패턴을 표현할 수가 있다. 그러나, 본 부분에서는 각 부분을 더욱 세밀히(숫자의 경우:10개, 알파벳의 경우 각각 13개씩) 할당할 이유는 각 패턴이 유효한 출력값을 하나만을 가짐으로 오인식이 발생할 경우에 이를 방지할 가능성을 더욱 높였다.

숫자의 경우 나타날 수 있는 출력값은 총 1024가지이고 이 중에서 유효한 출력값의 경우는 10개이므로 10개를 제외한 나머지 1014개의 값은 오류임을 알 수 있다. 알파벳의 경우도 마찬가지이다. 오류가 발생하게 되면 오류에 해당하는 문자를 인식하지 않고 "?"로 남겨둔 뒤, 후처리 과정(DB 매칭) 수행시에 이를 제외시킴으로써 매칭도를 높일 수 있다.

3.4 문자의 정형화

입력되는 각 문자는 보통 50*80 정도의 크기로 입력된다. 그러나 이것은 상황에 따라서 매우 유동적

이다. Scaling은 이러한 다양한 크기의 입력 패턴을 모두 같은 크기의 문자로 정형화하고 나아가서 각 문자가 갖고 있는 noise들을 적절히 감쇄시켜 인식을 또한 높일 수 있다.

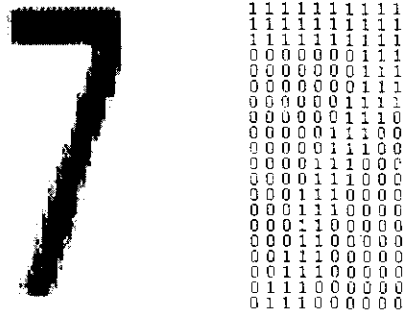
입력되는 각 문자는 10*20의 크기로 고정시키며 다음과 같은 방법으로 정형화 과정을 수행한다. 여기서 적용된 공식은 다음과 같다. 만약,

$$\text{영역 픽셀수} * 0.2 < \text{검정 픽셀수}$$

가 만족하면 Target(x,y) = 1(검정)

만족하지 않으면 Target(x,y) = 0(흰색)으로 처리한다.

Scaling 과정을 통해서 얻어진 컨테이너 문자는 다음과 같다.



IV. 실험 평가 및 결과

4.1 컨테이너 식별자 학습

본 부분의 BP망은 10*20의 정규화된 데이터할 가지고 학습을 시킨다. 학습을 시키기 위한 파라미터 값은 표 4.1과 같다.

	BP1	BP2	BP3
Training file	BP1.TRN	BP2.TRN	BP3.TRN
Network File	BP1.NET	BP2.NET	BP3.NET
Input Node	200	200	200
Middle Node	150	150	150
Output Node	10	13	13
Learning Rate	0.013	0.013	0.013
Tolerance	0.2	0.2	0.2
Momentum	0.9	0.9	0.9
Sample	243	138	169

표 4.1 뉴럴넷 문자인식의 파라미터 값

학습 패턴의 개수는 BP1은 243개, BP2는 138개, BP3는 169개로 총 550개의 학습 패턴으로 구성된다.

네트워크는 입력층(Input layer), 중간층(Hidden layer), 출력층(Output layer)등 총 3개의 계층으로 구성된다. 입력층은 200개의 노드로 구성이 되고 중

간층은 150개의 노드로 구성된다. 출력층의 경우는 BP1, BP2, BP3 각각 10, 13, 13의 노드로 구성된다.

출력 가중치를 생성하는 파일은 BP1.net, BP2.net, BP3.net으로 각각의 학습 시간은 BP1의 경우 2시간 24분, BP2의 경우 2시간 17분, BP3의 경우 2시간 32분의 시간이 소요된다.

5.2 컨테이너 식별자 인식

숫자의 경우는 BP1 네트워크망, 알파벳은 BP2, BP3 네트워크망을 통해 인식 결과를 얻는다. 인식의 결과로써 단일 인식, 중복 인식, 무인식의 경우로 나타날 수 있다.

- (1) 단일 인식 : 네트워크망을 통해 인식된 문자가 하나인 경우
- (2) 중복 인식 : 네트워크망을 통해 인식된 문자가 두 개 이상인 경우로 이때 인식된 문자들 중에서 가장 유사도가 높은 문자를 인식결과로 출력한다.
- (3) 무인식 : 네트워크망을 통해 인식된 문자가 하나도 없을 경우로 이때에는 출력 값으로 "?"를 출력한다

5.3 인식 결과의 후처리

후처리는 컨테이너 식별자 인식 결과를 확인하고 수정할 수 있도록 하는 부분이다. 이 부분에서 오인식된 문자인 "?"를 컨테이너 DB 매칭에서 제외시킨다. 이는 결과적으로 후처리의 수행 능률을 향상시키는 결과를 가져온다. 그 이유는 컨테이너 식별자 중에서 오인식된 문자를 결과 값으로 얻어 후처리를 수행하게 되면 잘못된 DB 매칭의 결과를 가져올 확률이 높아지기 때문이다.

V. 결 론

본 논문은 컨테이너 문자인식에 Back-propagation을 이용하여 인식률을 높이기 위한 알고리즘을 적용한 것이다.

기존의 컨테이너 문자 인식의 경우 하나의 컨테이너 식별자를 인식하는데 걸린 시간은 2~3초의 처리 시간이 소요되었으나, Back-propagation을 사용한 컨테이너 문자인식에서는 약 17초 정도의 시간이 소요되고 있다. 이렇게 인식 시간이 많이 걸리는 이유는 수행시마다 네트워크망을 메모리에 적재하는데 이로 인한 시간 손실이 크기 때문이다. 따라서 현재는 이를 보완하여 한번만 네트워크망을 메모리에 적재하여 수행할 수 있도록 하여, 인식하는데 걸리는 시간은 3~4초로 기존의 문자인식보다 약간의 시간이 더 소요된다. 하지만 인식률에서 더 좋은 결과를 보이고 있고, 컨테이너 처리에서 이 정도의 지연은 감수할 수 있을 정도의 시간이므로 그리 큰 영향은

미치지 않는다.

인식률의 경우 기존 컨테이너 인식의 경우 약 96%의 인식률을 보였으나 Back-propagation을 이용한 경우 이보다 향상된 98%의 인식률을 보이고 있다.

앞으로 알고리즘의 보완으로 인식하는데 걸리는 시간을 단축시키고 보다 많은 패턴을 학습시키면 더 나은 결과가 기대대리라 예상된다.

참고문헌

- [1] 손,尹, "신호처리 기법을 응용한 차량번호판 추출 방법," *전자공학회지*, vol. 30, no. 7, pp. 92~101, Jul. 1993.
- [2] Sing-Taz Bow, "Pattern Recognition and Image Preprocessing," Marcel Dekker, Inc. 1992
- [3] 이성근, "항만 물류처리 자동화를 위한 컨테이너 식별자 처리 및 인식에 관한 연구", 경원대학교 대학원 석사논문, 1997
- [4] Yuichi TOGASHI and Hideaki YAMAGUCHI, "Stable Position Measurement of a Moving Air plane by Adaptive Camera Control," MVA'94 IAPR Workshop on Machine Vision Application Dec. 13~15, 1994
- [5] 김 태균, "Syntactic法에 의한 한글의 패턴 인식에 관한 연구," *전자공학회지*, No.5, pp.154~160, 1977
- [6] Stephen P.Banks, "Signal Processing, Image Processing and Pattern Recognition," Prentice Hall, 1990
- [7] P.D.Wasserman, "Nerual Computing : Theory and Practice," Van Nostrand Reinhold. 1989
- [8] 이만형, 허도영, 이성근, 황대훈, "항만 물류처리 자동화를 위한 컨테이너 식별자의 영상 전처리에 관한 연구" 1998년 4월, 정보처리학회 춘계학술발표대회 논문지