

Matlab을 이용한 자동 점자 변환기

김현진* · 김예찬* · 박창진* · 오세종* · 이봉주**

Auto Braille Translator using Matlab

Hyun-JIn Kim* · Ye-Chan Kim* · Chang-Jin Park* · Se-Jong Oh* · Boong-Joo Lee**

요 약

본 논문은 시각 장애인들을 위해 영상처리 기반의 자동 점자 변환기의 설계 및 구현에 관한 내용을 기술한다. 영상처리 기반의 변환 알고리즘은 웹캠으로 획득한 입력 영상을 이진 영상화 한 다음, 문자 영역을 라벨링 처리하여 저장되어 있는 문자 패턴 영상과 상호 상관도를 계산하여 해당되는 점자로 변환한다. 컴퓨터 시뮬레이션을 통해 제안한 알고리즘을 모의 실험한 결과, A5 용지에 인쇄된 숫자와 알파벳에 대하여 각각 95%, 91% 변환 성공률을 보여 주었고, 아두이노를 이용하여 서보모터로 구현한 시제품 시험을 통해 89% 변환 성능을 확인함으로써 구현된 자동 점자 변환기의 실용화 가능성을 확인하였다

ABSTRACT

This paper describes the design and implementation of automatic braille converter based on image processing for a person who is visually impaired. The conversion algorithm based on the image processing converts the input image obtained by the web-cam to binary image, and then calculates the cross-correlation with the stored character pattern image by labeling the character area and converts the character pattern image into the corresponding braille. The computer simulations showed that the proposed algorithm showed 95% and 91% conversion success rates for numerals and alphabets printed on A5 paper. The prototype test implemented by the servo motor using Arduino confirmed 89%, conversion performance. Therefore, we confirmed the feasibility of the automatic braille transducer.

키워드

Matlab, Image Processing, Servo Motor, Webcam, Arduino Uno
매트랩, 영상 처리, 서보 모터, 웹캠, 아두이노 우노

1. 서 론

정보를 얻을 수 있는 방법이 늘어가고, 정보의 수준이 급격히 변화하고 있는 현대 사회에서 시각장애인에게 제공되고 있는 정보의 양은 굉장히 한정적이

다. 관공서나 행정기관에서 제공되는 안내책자만 보더라도 시각장애인을 위해 제공된 점자나 음성변환 서비스를 찾기 쉽지 않다. 또한 장애인단체가 주최하는 세미나만 가더라도 시각장애인이라고 미리 알리지 않으면 점자나 음성변환 서비스를 제공하지 않는 경우

* 남서울대학교 전자공학과 (ckdwls2525@naver.com, dhtpwhd3@naver.com, urususk@daum.net, aasin201@naver.com)

** 교신저자 : 남서울대학교 전자공학과

• 접수일 : 2017. 06. 16
• 수정완료일 : 2017. 07. 13
• 게재확정일 : 2017. 08. 01

• Received : Jun 16, 2017, Revised : July 13, 2017, Accepted : Aug 01, 2017

• Corresponding Author : Boong-Joo Lee
Dept. of Electronic Engineering, Namseoul University,
Email : bjlee@nsu.ac.kr

도 많은 상황이다.

이 상황에 대하여 장애인단체에서는 모든 자료에 점자 서비스를 제공하고 싶지만 생각보다 예산이 열악한 장애인단체로서는 따로 시각장애인 신청을 받지 않는 한 서비스를 제공하는 것이 쉽지 않다고 말하고 있다. 즉, 정부나 지자체가 시각장애인에게 직접적으로 관련하여 필요한 자료인지 비용이 얼마나 필요한지에 따라 판단하고 제공할 수 있는 수준의 서비스가 아직 준비되지 않았다는 것이다.

시각장애인을 위한 장애인 복지 법 제 22조에서 시각장애인 정보접근을 위해, 점자·음성 변환용 코드를 삽입된 자료를 국가가 제공할 것을 명시하고 있다. 또한 장애인이 원하는 정보를 얻고 이를 자기의 의사에 맞게 선택할 수 있도록 보장하는 규정이 장애인 차별금지법 제 7조에 명시적으로 규정하고 있다[1].

이 규정들은, 국가가 제공하는 모든 정보와 자료에 대해 시각장애인도 접근할 수 있도록 마련한 제도이다.

시각장애인들도 비장애인들과 마찬가지로, 정부나 지자체에서 나오는 정책이나 규제들을 미리 알아야 관련 사항을 접근할 수 있기 때문이다. 또한 정부와 지자체는 점자나 음성안내 서비스를 저렴한 비용에 보편적으로 쓰일 수 있도록 기술을 개발하여야 한다고 생각한다. 이는 배려나 시혜가 아닌 의무이며, 국민과의 대화이자 소통의 도구이다.

특히, 시각 장애인이 세상과 소통할 수 있는 것은 각종 도서라고 할 수 있는데 시각 장애인들이 이용할 수 있는 점자도서는 그 종류가 한정적이며, 제작에도 어려움이 많아 다양한 도서를 시각 장애인에게 제공해 주지 못하고 있다. 기존의 시각 장애인들을 위한 문자점자 변환은 타자기 형태의 수동형 변환기를 이용하여 제한된 점자 서적으로 인쇄하는 방식이다.

따라서 본 논문에서는 시각장애인들에게 보다 폭넓은 자료들을 읽을 수 있도록 화상으로 입력된 영상을 처리하여 문자를 실시간으로 변환하게끔 하는 자동 점자 변환 시스템에 대하여 연구하였고 이에 대한 최적화된 시작품을 제작하였다.

컴퓨터와 연결된 웹캠으로 획득한 영상을 처리하여 점자로 변환하는 출력시키는 알고리즘의 순서도는 그림 1과 같으며 동작 설명은 다음과 같다.

웹캠으로 영상을 획득한 후, RGB 영상을 그레이스케일 영상으로 변환한다. 그레이스케일 영상은 이진영

상으로 변환되고 이진 영상의 문자 영역은 라벨링 동작을 수행하여 문자 형태로 그룹핑 한다[2]. 설정된 위치정보를 이용하여 분할된 문자 영역은 데이터베이스화된 샘플영상과 상호상관도를 계산하여 최대값을 가지는 숫자를 탐색하여 해당 점자로 출력하는 동작을 반복 처리한다.

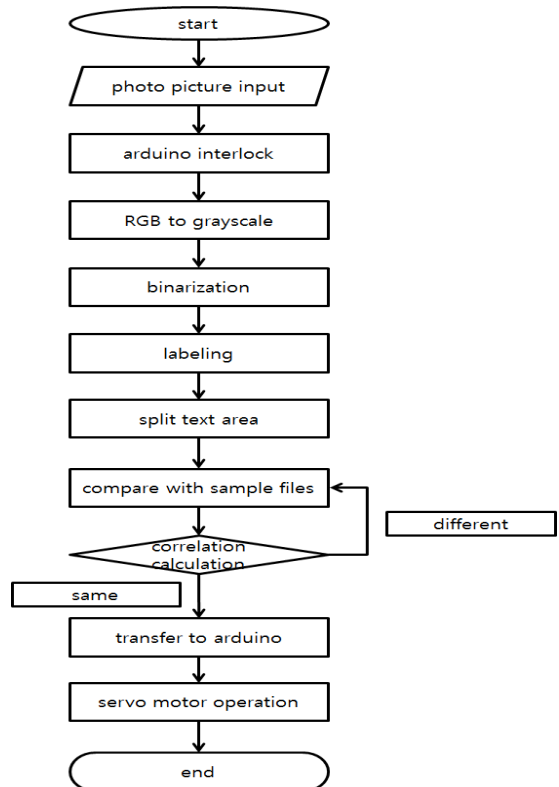


그림 1. 플로우 차트
Fig. 1 Flow chart

II. 시스템 구성

2.1.1 사진 영상 입력

웹캠으로 찍은 영상을 Matlab에서 영상처리하기 위해 촬영한 사진 중 가장 최근에 찍힌 파일을 읽어와 한다. 그러기 위해 다음과 같은 작업을 수행한다.

먼저 사진폴더가 있는 경로를 Matlab을 이용해 임의의 변수에 저장한다. 그 다음 변수를 이용하여 현재 작업 폴더 외부에 있는 경로의 사진파일을 읽고 그중에 가장 최근에 촬영된 파일을 원본파일로 설정한다.

2.1.2 RGB to 그레이 스케일

웹캠을 통해 입력된 영상은 RGB 포맷으로 획득되기 때문에 영상처리를 위하여 명암도 영상으로 변환한다. 변환식은 식(1)을 사용한다. 식(1)을 통해 숫자 4라는 영상파일에 그레이스케일 작업을 실행한 결과는 그림 2과 같다.

$$Y_{(x,y)} = 0.2126R_{(x,y)} + 0.7152G_{(x,y)} + 0.0722B_{(x,y)} \quad (1)$$

251	251	251	255	209	34	31	230	254	251
251	251	255	225	44	0	15	221	255	251
251	255	240	62	34	71	10	221	255	251
253	247	86	9	194	119	6	221	255	251
255	114	1	162	255	114	6	222	255	251
138	1	133	255	255	110	5	220	255	252
19	0	66	79	63	16	0	46	90	245
200	179	193	202	191	80	3	158	204	253
255	255	245	162	157	64	0	116	144	201
251	253	237	97	85	102	106	92	66	166

그림 2. 그레이스케일 이후 영상파일의 행렬
Fig. 2 Matrix of image file after grayscale

2.1.3 이진화

이진화란 원래 엄밀한 의미로서는 그 값을 0 또는 1로 표현하는 것이다. 그러나 영상의 픽셀 값은 흑과 백으로 이루어진 0~255범위의 값으로서, 0 또는 1로 표현할 경우 눈으로 보기에 식별이 잘 안되기 때문에 영상의 이진화에서는 픽셀값을 0 또는 255로 변환한다.

이진화를 수행하기 위해서는 영상 내 모든 픽셀에 대하여 그레이스케일 값이 특정 값보다 크면 흰색을 나타내는 255로 바꾸고, 작으면 흑백을 나타내는 0으로 바꾸는 방법을 사용한다[3]. 이때, 픽셀 값의 크기를 비교하는 대상이 되는 값을 임계값이라 한다. 임계값은 그레이스케일 범위인 0 ~ 255 사이의 정수 값을 사용하며, 이진화 과정을 수식으로 표현하면 다음과 같다.

수식에 따라서 Matlab 과정을 통해 숫자 4라는 영상파일을 이진화 작업을 한 결과는 그림 3과 같다.

$$B(x,y) = \begin{cases} 255 & \text{if}(D(x,y) > \text{threshold}) \\ 0 & \text{else} \end{cases} \quad (2)$$

1	1	1	1	1	0	0	1	1	1
1	1	1	1	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	0	1	0	0	1	1	1
1	0	0	0	1	0	0	1	1	1
0	0	0	1	1	0	0	1	1	1
0	0	0	0	0	0	0	0	0	1
1	1	1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0

그림 3. 이진화 이후 영상파일의 행렬
Fig. 3 Matrix of image file after binarization

이처럼 영상의 이진화는 다양한 영상 처리 분야에서 사용되며, 특히 영상 내에 원하는 객체의 위치를 찾기 위한 전처리 과정으로 많이 사용된다.

2.1.4 라벨링

라벨링이란 인접하게 연결되어 있는 모든 픽셀을 하나의 개체로서 인지하여 동일한 번호를 붙이고(라벨을 붙이는 작업) 다른 연결 성분에는 또 다른 객체 번호를 붙이는 작업으로써, 이진화 처리하여 0과 255 값만으로 만든 흑백영상을 가지고 라벨링 작업을 수행하게 된다. 영상은 라벨링 명령을 통해 서로 연결된 인접 픽셀들의 값이 동일한 하나의 번호로 지정되어 나타나며 다른 인접영역 성분은 또 다른 번호로 지정하여 라벨링 된 영상을 지정된 특정번호로 쉽게 인식할 수 있도록 해준다. 이를 통해서 원래 입력한 영상에서 여러 개의 물체영역으로 분리하고 각각의 물체를 임의의 한 라벨 번호를 가진 영상으로 추출하여 각각 독립적으로 부를 수 있는 상태가 된다[4].

이진화 영상을 통해 라벨링을 하게 되면 더욱 흑과 백의 구분이 명확하기 이루어져 더욱 쉬운 라벨링 작업을 실행할 수 있기 때문에 이진화를 수행한 후에 라벨링을 하는 것이 일반적이다.

그림 4는 이진화된 숫자 값이고 그림 5는 그림 4에서 이진화된 영상을 각각의 개체로 라벨링 한 것이고, 이를 명확하게 보기 위해 구분해놓은 사진이다.

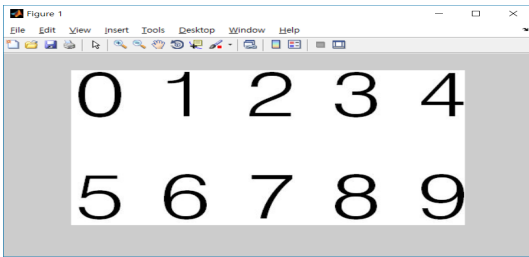


그림 4. 입력 영상 파일
Fig. 4 Input image file

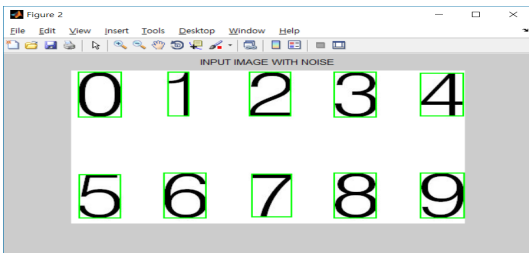


그림 5. 인접 영역에 대하여 라벨링
Fig. 5 Labeling for adjacent areas

2.1.5 글자 영역 분할

이진화 된 영상 또는 라벨링 된 영상에서 영역의 분할을 하려면 일정한 밝기 값을 가지는 영역의 경계를 추적하여 경계 픽셀의 순서화된 정보를 얻어내는 것이다.



그림 6. 라벨링 후 글자영역 분할 모습
Fig. 6 Appearance of letter region after labeling

라벨링을 통해 지정된 번호를 추출하기 위해서는 경계를 추적하는 작업이 필요하다. 경계를 추적하기 위해서는 1픽셀 두께를 가지는 픽셀의 순서화된 연속 체인 정보를 얻어야 한다. 그 영상에 대해 밝기 값 255를 가지는 영역을 기준으로 영역 안의 라벨링 된 영

상을 각각 추적하는 것이 가능하게 된다. 그림 6은 그림 5를 통해 각각 라벨링된 영상을 나열한 그림이다.

2.1.6 샘플영상과 비교

문자를 라벨링한 후 분할된 글자영역과 가장 유사한 문자로 표출하기 위해 그림 7과 같은 샘플영상들을 미리 만들어두어 읽어 들인 문자와 비교하도록 하였다.

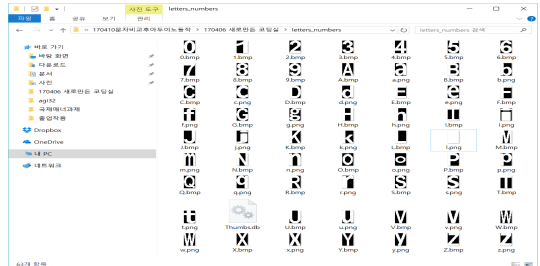


그림 7. 문자 비교를 위한 샘플영상
Fig. 7 Sample images for character comparison

2.1.7 상관도 계산

샘플영상과 라벨링 된 문자의 상관관계를 비교하기 위하여 먼저 두 배열의 크기를 맞춰주기 위해 리사이징을 해준다. 리사이징 과정은 샘플영상과 입력영상을 비교하기 위해서 반드시 필요한 과정이다. 영상의 사이즈가 다를 경우 비교에 문제가 되기 때문에 임의로 사이즈를 맞춰주게 된다.

그 후 행렬의 열을 더해주어 그림 8과 같이 1차원 히스토그램을 만들어 입력된 영상과 비교하게 된다. 두 행렬을 빼주어 비교 시 일정 수치 범위 안에 값이 존재한다면 같은 영상이라고 판별하게 설정하였다[5].

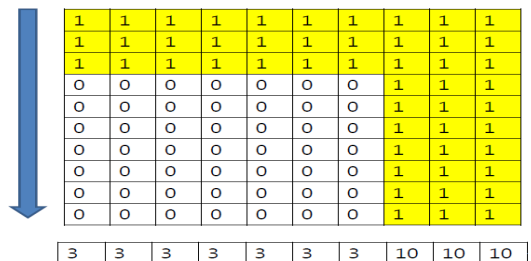


그림 8. 1차원 히스토그램 만들기
Fig. 8 Create a one-dimensional histogram

이미지의 크기를 설정해주었다면, 히스토그램을 만들어 구해진 샘플들의 히스토그램과 입력된 영상의 히스토그램을 비교한다. 그 과정은 그림 9와 같이 전체 샘플 히스토그램에서 하나의 입력 히스토그램을 반복 비교를 통해 행렬을 뺄셈 연산하였을 시 그 값 중 가장 유사한 문자를 찾아 문자가 동일하다는 판별을 하게 된다.

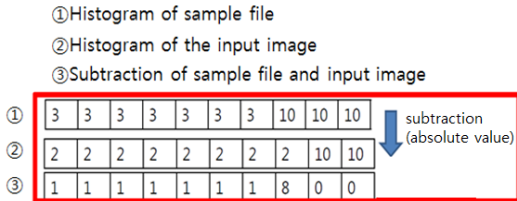


그림 9. 히스토그램을 통한 판별과정
 Fig. 9 Histogram identification process

2.2 서보모터 제어

Matlab을 통한 영상처리를 마친 이후 영상들의 상관관계를 비교하여 문자를 인식하게 되고, 그 값을 기반으로 하드웨어로 점자를 표현하기 위하여 아두이노 서보모터를 제어하여 표출하도록 해주었다. 그 과정에서 시각장애인이 명확히 인지하고 넘어갈 수 있도록 서보모터의 순차적인 움직임을 제어하기 위해 버튼을 이용하였다.

2.2.1 아두이노 연동

Matlab과 아두이노를 동시에 사용하려던 와중 Matlab 상에서 아두이노와 연동할 수 있는 지원 패키지를 알게되었고, 아두이노 코딩을 기반으로 Matlab 상에서 코딩하여 아두이노를 제어하기로 하였다[6].

Matlab상에서 아두이노를 사용할 수 있게끔 연동시켜주는 파일과 아두이노로 업로드하게 만들어 주는 파일을 활용하여, 위 파일들을 기반으로 아두이노 상에서 스케치 파일을 업로드 시켜주면 Matlab과 연동이 가능해진다.

III. 점자표 기반 하드웨어 제작

3.1 점자표지 표준규격

위처럼 영상을 통한 문자인식이 완료 되었다면, 이제 우리가 사용하는 숫자나 알파벳을 시각장애인이 읽을 수 있는 점자로 표현하여야한다.

우선적으로 점자 핀에 대한 표준규격은 그림 10와 같다. 표준규격을 통해 서보모터의 배치와 돌출 높이를 조절하여 시각장애인이 점자를 충분히 인식할 수 있도록 하드웨어를 제작하였다.

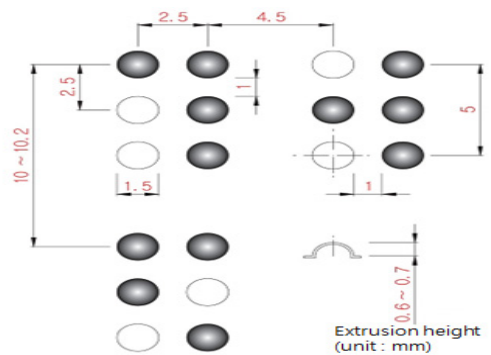


그림 10. 기술표준원에서 지정한 점자표지표준
 Fig. 10 Braille mark standard specified by the Korea Standards Institution

3.2 하드웨어 구현

제안하는 영상처리 기반 문자-점자 변환기의 전체 기능 구성도는 그림 11와 같다. 웹캠은 영상 획득 기능을 수행하고 Matlab을 통해 영상처리 작업을 수행하여 상관관계를 판별하고, 그 결과 값을 아두이노를 통해 직접 모터제어를 이용하여 점자를 표시하는 기능을 수행한다[7].

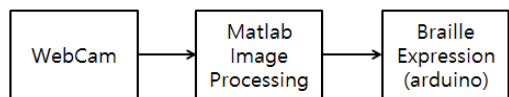


그림 11. 영상처리 기반 점자 변환 과정
 Fig. 11 Image processing based braille conversion process

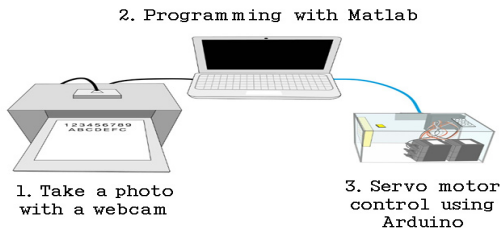


그림 12. 구현된 문자-점자 자동 변환기
Fig. 12 Implemented character-to-braille converter

웹캠을 사용하여 변환을 위한 영상사진을 촬영하였으며, 영상처리를 위하여 Arduino Uno를 사용하였다.

위 과정을 통해 구현된 전체 시스템 구상도는 그림 12과 같다. 웹캠과 영상처리 보드는 컴퓨터를 통해 데이터를 교환하고, 카메라의 높이는 47cm로 고정시켜 A5 가로 크기의 용지전체가 영상에 담길 수 있도록 설계해주었다. 점자 표시부는 서보모터의 회전을 이용하여 핀을 올려주는 형태로 구현하였다.

IV. 실험 및 고찰

4.1 히스토그램의 오차율

1차원 행렬을 통해 히스토그램을 만들어 문자비교를 시행하려 하였으나 그림 13과 같이 숫자 5, 6, 7, 9가 다른 숫자와 동일하다고 판별이 되어 인식률이 60%가 되므로 좋지 않은 판별능력이라고 생각되어 2차원 행렬을 통해 히스토그램을 만들기로 하였다[8].

sample \ input	0	1	2	3	4	5	6	7	8	9
0	8	40	16	14	30	13	8	18	14	8
1	44	2	32	36	32	37	42	26	40	40
2	19	31	5	11	19	10	17	9	13	17
3	16	36	10	6	26	9	12	16	12	12
4	35	27	25	25	11	30	33	21	35	29
5	17	31	5	13	15	12	15	7	13	15
6	13	37	13	11	23	16	11	15	19	15
7	27	23	17	21	19	22	27	15	25	27
8	12	38	6	12	20	13	10	10	8	8
9	15	39	9	7	25	10	9	15	11	9

setting standard : resize matrix - 10 x 10 / font size - 70p

그림 13. 숫자에 대한 히스토그램 비교
Fig. 13 Histogram comparison for numbers

4.2 숫자와 알파벳의 구분

숫자와 알파벳을 동시에 점자 핀으로 표출하려던 과정에서 표 1과 같이 일부 문자에 한해 혼동이 일어나 인식하는데 오류가 발생하였다. 따라서 숫자와 알파벳 구분을 명확히 해주기 위해서 숫자 변환과 알파벳 변환 과정을 나누어 주기로 하였다.

표 1. 숫자별 혼동되는 알파벳
Table. 1 Alphabet confused by number

0	1	2	5	6	8	9
O, D	I, T	Z	S	B	B	D

4.3 글씨 크기에 따른 문자 인식률

숫자와 알파벳에 대해 임의의 크기 10, 20, 30, 40, 50p에 대한 순서로 인식률을 실험해 본 결과 10p일 때는 인식이 잘 되지 않았으며, 30p일 때 평균 87%로 가장 인식률이 좋다고 판단하여 선택하기로 하였다.

그림 14, 16는 글씨크기에 따라 변환된 텍스트(숫자, 알파벳)를 나타낸 것이고, 이를 토대로 5차례의 실험을 통한 인식률을 나타낸 그래프가 그림 15, 17이다. 그 결과 숫자 크기 30p 이상일 시에 84% 이상의 인식률을 보여주며, 알파벳 크기 20p 이상일 시에 89% 이상의 인식률을 보여준다. 숫자와 알파벳에 대해 보편적인 인식률인 30p를 기본 문자 크기로 지정해 주었다.

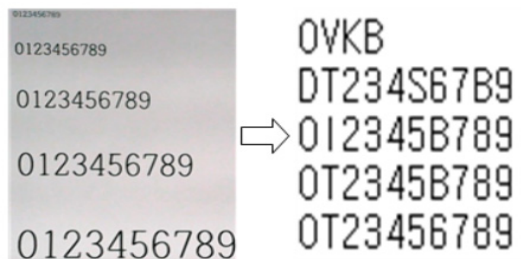


그림 14. 글씨 크기에 따른 숫자 변환
Fig. 14 Numeric conversion by font size

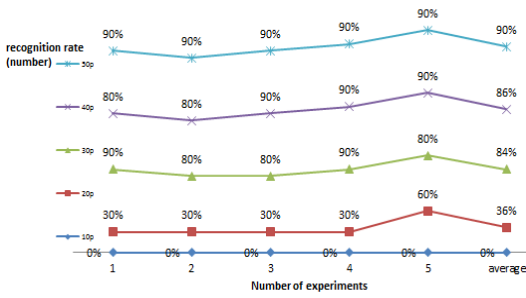


그림 15. 실험 횟수에 따른 인식률(숫자)
Fig. 15 Recognition rate according to experiment frequency(number)

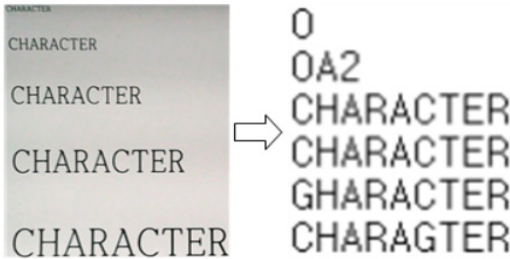


그림 16. 글씨 크기에 따른 알파벳 변환
Fig. 16 Alphanumeric conversion by font size

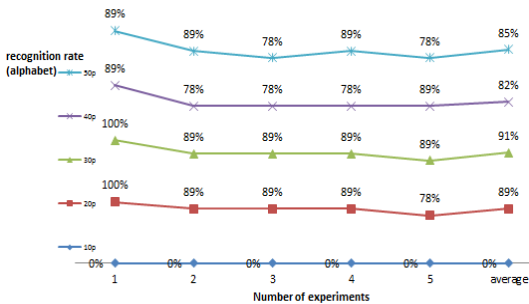


그림 17. 실험 횟수에 따른 인식률(알파벳)
Fig. 17 Recognition rate according to experiment frequency(alphabet)

4.4 글씨체에 따른 문자 인식률

위 실험결과에 따라 30p를 기준으로 숫자와 알파벳에 대해 임의의 글씨체 함초롱바탕, 궁서, 굴림, 중고딕, 신명조 순으로 인식률을 실험해 본 결과 숫자에 대해선 궁서체가 94%로 가장 인식률이 좋았고, 알파벳은 함초롱바탕이 96%로 가장 인식률이 좋았다.

그림 18, 20은 글씨체에 따라 변환된 텍스트를 나타낸 것이고, 이를 토대로 5차례의 실험을 통한 인식률을 나타낸 그래프가 그림 19, 21이다.

그 결과 숫자일 경우에 중고딕, 알파벳일 경우에 함초롱바탕이 평균 91% 이상의 인식률을 보여주기 때문에 함초롱바탕을 기본글씨체로 지정해 주었다.

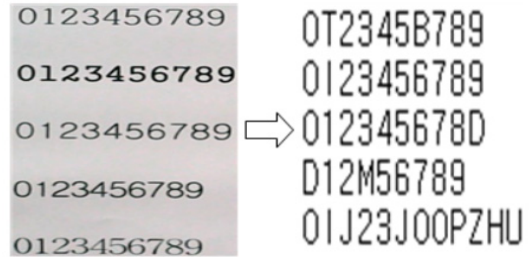


그림 18. 글씨체에 따른 숫자 변환
Fig. 18 Numeric conversion by font

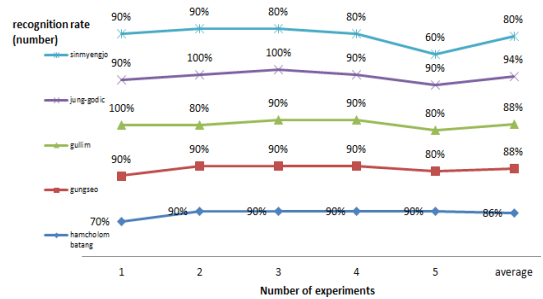


그림 19. 실험 횟수에 따른 인식률(숫자)
Fig. 19 Recognition rate according to experiment frequency(number)

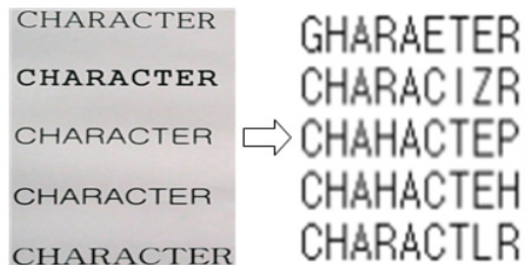


그림 20. 글씨체에 따른 알파벳 변환
Fig. 20 Alphanumeric conversion by font

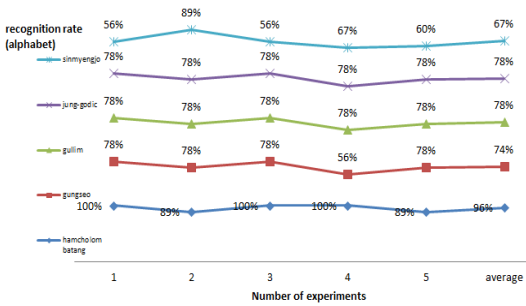


그림 21. 실험 횟수에 따른 인식률(알파벳)
Fig. 21 Recognition rate according to experiment frequency(alphabet)

4.5 카메라 성능에 따른 행렬의 차이

Matlab 명령어중 imread 명령어로 영상을 받아들이었을 시 카메라의 성능에 따라 각 행렬(픽셀)의 값이 다르다는 것을 그림 22과 같이 확인하였다. 이는 해상도에 따라 픽셀의 개수가 다르기 때문인 것으로 확인되었고, 이에 따라 좋은 성능의 카메라를 사용한다면 보다 자세한 영상처리를 할 수 있을 것이다.

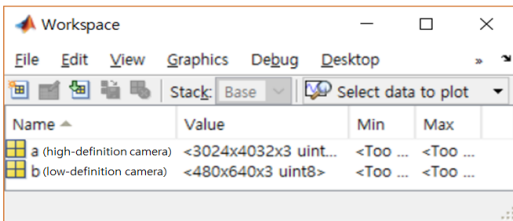


그림 22. 카메라에 따른 행렬의 값
Fig. 22 The value of the matrix according to the camera

V. 결론

본 논문에서는 앞을 볼 수 없는 시각장애인들에게 보다 나은 질의 정보를 제공할 수 있도록 Matlab을 통한 영상처리 후 아두이노 모듈을 기반으로 점자 판으로 제어할 수 있도록 구현하였다. 입력된 영상에서 문자영역을 분할하고, 샘플로 저장된 참조 문자패턴과 상호상관도를 계산하여 가장 유사한 문자를 점자로 변환하는 알고리즘을 적용하였다. 서보모터 작동 결과

94%의 문자 변환 성능을 보여주었다. 이를 통하여 Matlab 영상처리 기반의 자동 점자 변환기의 실용화 가능성을 확인하였으며, 향후 한글 변환 과정에 대하여 성능을 개선함으로써 시각장애인들의 복지 증진에 기여할 수 있을 것으로 보인다.

부족했던 점 중에 가장 큰 것은 한글 인식이다. 일반적으로 영어나 숫자 같은 경우에는 각각 하나의 단어가 독립적인 개체로 이루어져 따로 구분하기 가능하나, 한글의 경우 초성, 중성, 종성의 각각의 단어가 하나의 글자이고 샘플파일을 설정할 경우, 글자의 경우의 수가 24,389가지가 나오기 때문에 샘플 파일을 만드는 것과 그것을 비교하여 하나의 글자를 표출하는 것이 어려워 시간적인 측면에서 한글은 따로 설정하지 못하였다. 추후에는 한글에 대한 구분 또한 이루어지길 바라는 마음이다.

References

- [1] J. Chae, W. Kim, "Implementation of DSP Embedded Numerical - Braille Transform Image Processing Algorithm," *J. of the Korea Communications Satellite Industrial Research Society Foundation*, vol. 11, no. 2, 2016, pp. 477-478.
- [2] D. Kim and E. Cha, "A Method for Binarization and Stroke Reconstruction of Low-Quality Character Images for Effective Character Recognition," *J. of the Korea Communications Satellite Industrial Research Society Foundation*, vol. 11, no. 3, 2007, pp. 608-618.
- [3] Rafael C, Gonzalez, Richard. Woods, Steven L, and Eddins. *Digital image processing using MATLAB*. McGraw-Hill, 2011.
- [4] A. McAndrw, *Introduction to digital image processing with MATLAB*. Alasdair McAndrew, 2004.
- [5] W. Oh and S. Lee, "An Effective Algorithm for Diagnosing Sensor Node Faults," *J. of the Korean Institute of Electronic Communication Sciences*, vol. 10, no. 2, 2015, pp. 283-288.
- [6] S. Lee and D. Park, "Acoustic Event Detection-Reaction System Based on Arduino-MATLAB/Simulink Interoperation Environment," *J. of the Korean Institute of*

Electronics and Information Engineers, vol. 2015, no. 6, 2015, pp. 1537-1540.

- [7] Y. Park, "Development of Smart laser Pointer using Image Processing," *J. of the Korean Institute of Electronic Communication Sciences*, vol. 11, no. 12, 2016, pp. 1245-1250.
- [8] B. Kim, "Algorithm to Apply Numerical Information based on Mnemonic System," *J. of the Korean Institute of Electronic Communication Sciences*, vol. 10, no. 6, 2015, pp. 677-682.

저자 소개



김현진(Hyun-Jin Kim)

2017년 남서울대학교 전자공학과
졸업(공학사)

※ 관심분야 : 통신 시스템, 디스플레이 공학



김예찬(Ye-Chan Kim)

2017년 남서울대학교 전자공학과
졸업(공학사)

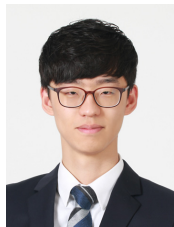
※ 관심분야 : 제어 시스템, 반도체 공학



박창진(Chang-Jin Park)

2017년 남서울대학교 전자공학과
졸업(공학사)

※ 관심분야 : 디스플레이 공학, 통신 시스템



오세종(Se-Jong Oh)

2017년 남서울대학교 전자공학과
졸업(공학사)

※ 관심분야 : 반도체 공학, 디스플레이 공학



이봉주(Boong-Joo Lee)

1996년 2월 인하대학교 전기공학과
졸업(공학사)

1998년 2월 인하대학교 대학원 전기공학과
졸업(공학석사)

2003년 2월 인하대학교 대학원 전기공학과 졸업(공학박사)

2007년 8월 LG전자 디지털 디스플레이 연구소
남서울대학교 전자공학과 부교수

※ 주 관심분야 : OLED, LED, 광소자, Solar-cell

