

시각장애인을 위한 라즈베리파이 기반 스마트 점자 변환기

홍석진, 정민수, 김수아, 정의림*

한밭대학교

lf2net89@naver.com, jmk3012@naver.com, twghub@naver.com, *erjeong@hanbat.ac.kr

Raspberry pi based Smart braille converter for visally impaired

Hong Seok Jin, Jeong Min Su, Kim Su A, Jeong Eui Rim*(Corresponding author)

Hanbat National Univ.

요약

본 논문은 스마트 점자 변환기의 설계한다. 시각장애인은 점자로 표시되지 않은 글씨를 읽을 수 없어 일상생활에 큰 어려움을 겪는다. 이를 해결하기 위해 읽고자 하는 글자를 촬영하면 자동으로 인식하여 점자로 변환해 출력해주는 스마트 점자 변환기를 설계한다. 라즈베리파이 4의 Linux 환경을 기반으로 사용 언어는 Python이다. 라즈베리파이 4 전용 카메라로 문자를 촬영하고, Python 패키지인 Easy OCR로 문자를 인식한 뒤, Python 라이브러리 hbcvt를 통해 점자로 변환한다. 점자 데이터는 라즈베리파이의 데이터 입출력을 담당하는 GPIO 핀 6개를 이용해 전자석의 원리로 설계된 하드웨어로 전달되어 실제 시각장애인이 읽을 수 있는 점자 형태로 나타난다.

I. 서론

스마트시대인 최근, 장애인들의 삶의 질을 높이는 시스템은 많이 개발되고 있지만, 장애인의 편의를 위한 시스템은 개발되는 수가 적다. 특히나 장애인들이 점자가 표시되어 있지 않은 표지판이나 상품 라벨을 읽는 방법은 타인의 도움 말고는 없다. 따라서 본 논문에서는 라즈베리파이 4를 이용해 점자로 표시되어 있지 않은 글자를 읽을 수 있는 스마트 점자 변환기를 설계한다.

라즈베리파이(Raspberry Pi)는 모니터와 키보드, 마우스 등과 같은 주변기기 연결이 자유로운 소형 컴퓨터이며, 간단하게 연결할 수 있는 전용 카메라가 존재한다. 아두이노(Arduino)는 외부 기기 제어에 특화되어 있어 변환 소프트웨어를 자체적으로 실행한 뒤 점자로 출력해야 하는 스마트 점자 변환기의 목적에는 라즈베리파이가 적합하다.

광학 문자 인식(Optical Character Recognition, OCR)은 사람이 쓰거나 기계로 인쇄한 문자의 영상을 기계가 읽을 수 있는 문자로 이미지 스캐닝하는 기술이다. 본 논문의 점자 변환기를 설계하는 데에는 Python의 패키지인 Easy OCR을 사용한다. Easy OCR은 문자 영역 인식(Detection)과 문자 인식(Recognition)의 정확도가 높다. 또한 구현이 간단하고, 100개 이상의 언어를 지원하고 있으며 꾸준히 업데이트되고 있다.

스마트 점자 변환기는 다음과 같은 과정으로 작동한다. 먼저, 라즈베리파이 4 전용 카메라로 문자를 촬영하고, Python의 Easy OCR 패키지를 활용하여 촬영한 사진 데이터에서 문자를 자동으로 인식해 추출한다. 추출된 문자는 Python의 hbcvt 라이브러리를 통해 자음과 모음을 분리 후 한글 점자로 변환된다. 한글 점자로 변환된 데이터는 전자석 점자 출력기를 통해 시각장애인이 읽을 수 있는 6점 점자의 형태로 출력된다.

II. 카메라 연결 및 문자 촬영

그림 1의 (a)는 라즈베리파이 4 모델 B로 1.5GHz의 CPU와 8GB의 메모리(RAM)를 갖추고 있다. 전원 공급부는 Type-C 이며, 운영체제는 자체 개발 Linux 기반 운영체제인 Raspberry Pi OS 64bit를 사용한다.

그림 1의 (b)는 라즈베리파이 4 전용 카메라로, 그림 1의 (a)에서 빨간 상자 부분의 포트를 통해 라즈베리파이와 연결한다. 'libcamera'라는 애플리케이션을 Linux 프롬프트에서 제어해줌으로써 사진을 촬영해 저장한다. 실제 운용할 Python 코드에서는 프롬프트에 해당 명령어를 입력해주는 os 명령어로 카메라에 촬영을 지시한다.

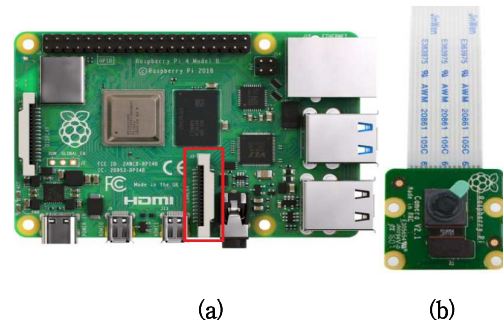


그림 1 라즈베리파이 4와 전용 카메라

III. 촬영한 문자의 점자 변환

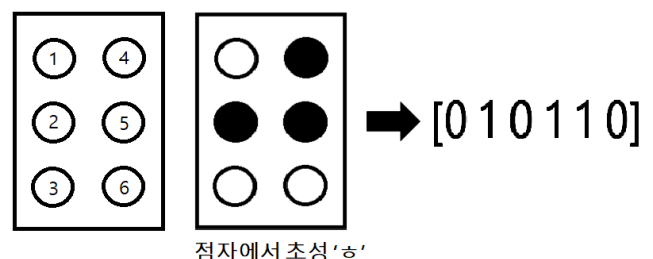


그림 2 초성 'ㅎ'의 점자 행렬 표현

점자는 일반적으로 6개의 점이 블록한가, 블록하지 않은가로 문자를 표기한다. 그림으로 표현 시에는 검은색 원이 블록한 점이고, 흰색 원은 블록하지 않은 점이다. 왼쪽 위 끝점부터 1번 점자라는 의미로 '1점'으로 부르며, 아래로 내려가면서 순서대로 '2점'부터 '6점' 순으로 번호를 붙인다. 이를 통해 그림2와 같이 행렬로 점자를 표현할 수 있다.

<촬영 사진>

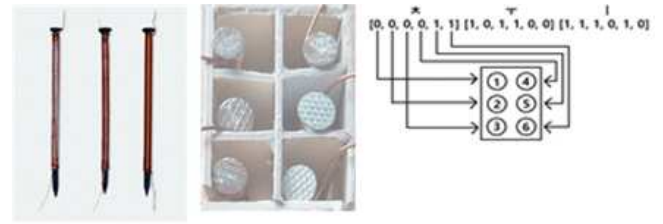


그림 3 촬영 사진의 글자 점자 행렬 변환

<출력 점자 행렬>

['취업 전략과', '실전취업']
[취
[大, [0, 0, 0, 0, 1, 1]]
[ㅌ, [1, 0, 1, 1, 0, 0]]
[업
[ㅇ, [1, 1, 0, 1, 1, 0]]
[ㅣ, [0, 1, 1, 1, 0, 0]]
[ㅂ, [1, 1, 0, 0, 0, 0]]
...

그림 5 점자 출력 하드웨어

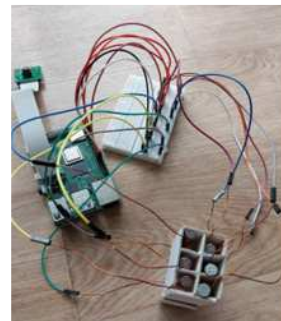


점자 출력을 담당할 하드웨어는 6개의 자석과 에나멜선을 감은 6개의 못으로 구성한다. 못에 감긴 에나멜선에 전류가 흐르면, 못이 전자석이 되어 자석의 성질을 띠게 된다. 못 아래에 같은 극이 마주 보게 자석을 배치해 두면, 척력이 작용하여 전류가 흐르는 못은 떠오르고, 흐르지 않는 못은 떠오르지 않는다. 점자 행렬의 0에 해당하는 GPIO 핀에는 전류가 흐르지 않고, 1에 해당하는 GPIO 핀에는 전류가 흐르게 하여 점자를 표현한다. 떠 있는 못은 점자의 블록한 점을 나타내고, 떠 있지 않은 못은 평평한 점을 나타낸다.

Python에서 Easy OCR을 호출한 뒤, easyocr. Reader 명령어에서 ['KO'](한국어)로 설정해주면 지정된 사진의 글자를 한국어로 인식하여 추출해 리스트의 형태로 저장한다. 그리고 join 명령어를 통해 리스트 형태의 문자를 문자열 형태로 변환시킨 뒤, hbcvt라는 점자 변환 python 라이브러리로 그림3과같이 문자의 초성, 중성, 종성을 구분해 한글 6점 점자 행렬로 바꾼다.

VI. 점자 출력

출력된 결과에서 불필요한 문자는 제외하고 점자에 해당하는 행렬 부분만 추출한 뒤, 하나의 자음 또는 모음이 되도록 6개씩 나누어서 저장한다.



pin0is high
pin1is high
pin2is low
pin3is high
pin4is high
pin5is low
pin0is low
pin1is high
pin2is high
pin3is high
pin4is low
pin5is low
pin0is high
pin1is high
pin2is low
pin3is low
pin4is low
pin5is low

그림 6 전체 모습 및 출력 검증

GPIO 1번은 점자 출력기의 1번 못과 연결하고, 2번은 2번끼리 연결해서 6개의 GPIO 핀이 같은 번호의 점자 출력기 못과 연결되도록 한다. 떠오르는 핀은 출력장에서 high 상태로 표시되고, 떠오르지 않는 핀은 low 상태로 표시된다.

V. 결론

본 논문에서는 카메라로 찍은 사진에서 문자를 인식, 점자로 변환 및 출력하는 스마트 점자 변환기를 제작했다. GPIO 핀 검증을 통해 에나멜선에 정상적으로 전류가 흐르는 것을 확인했지만, 못이 충분히 떠오르지 않아 시각장애인이 쉽게 읽기에는 어려움이 있었다. 못을 충분히 떠오르게 할 전압이 부족했던 것을 원인으로 분석했다. 추후 연구에서 추가전압 공급을 통해 전자석 세기를 강화하고 휴대성을 높이기 위해 부피를 줄이는 것으로 전체적인 성능을 개선할 수 있을 것이다.

참 고 문 헌

- [1] Yu Jin Choi, Jin Woo Kim, Ji Yong Park, "Translation System of Korean into Braille based on OCR for the Blind", 2021년도 한국통신학회 추계종합학술발표회, Nov. 2021
- [2] Hyo Hyun Choi*, Su Hyun Moon, "An Improvement Method for the Braille Labeling of Beverage Products Using OpenCV", 한국컴퓨터정보학회 하계학술대회 논문집 제30권 제2호, July, 2022

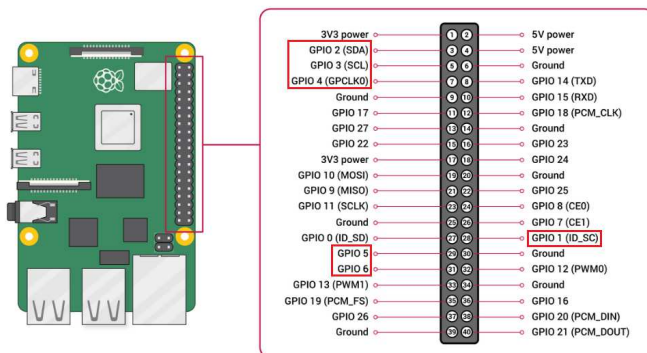


그림 4 라즈베리파이 4 GPIO 핀 배열

6개씩 나누어진 점자 행렬을 순서대로 GPIO 핀 1번부터 6번 핀까지 점자의 1점부터 순서대로 대응시킨 뒤, 6개씩 동시에 출력하도록 설정함으로써 실제 6점 점자의 모습을 표현한다. 각 점자 간 출력 간격은, 0.7초로 설정하여 순차적으로 점자를 읽는 데 문제가 없게 한다.