

차량 사고 감지 및 관리 시스템

정현진, 김민중, 이승환, 황소영*
부산가톨릭대학교

{jhj6740, leftkmj1362, leesh86030}@naver.com, *soyoung@cup.ac.kr

Vehicle Accident Detection and Management System

Hyunjin Jeong, Minjong Kim, Seunghwan Lee, Soyoung Hwang*
Catholic University of Pusan

요약

최근 뺑소니 사고의 검거율은 발전하는 CCTV와 블랙박스로 개선되고 있으나 사고 후 미비한 대처에 의해 미해결 사고가 매년 적게는 100여 건, 많게는 900여 건 발생하고 있다. 피해자의 의식불명 등 여러 이유로 사고 위치를 정확히 기억하지 못하거나 증거를 수집하지 못하여 사건이 미해결로 남는 경우도 있다. 본 논문에서는 차량의 사고를 감지하고 관련 정보를 수집, 저장, 관리하여 사후 처리에 대한 지원이 가능한 시스템을 제안한다. 제안하는 시스템은 사고 감지를 위한 센서부, 데이터 수집과 처리를 위한 모바일 장치 및 사고 정보 저장을 위한 클라우드 저장소로 구성되며 프로토타입의 구현 및 실험 결과를 제시한다.

I. 서론

우리나라의 뺑소니 사고 검거율은 90년대 초 40-50%에서 경찰의 단속, 블랙박스 및 CCTV의 대중화로 최근에는 평균 90% 이상으로 개선되었다. 하지만 미비한 사후대처에 의한 미해결 사고는 매년 발생하고 있는 실정이다[1]. 또한 미국에서는 2015년 737,100건의 뺑소니 사고가 발생했으며, 2016년에는 2,049명의 사망자가 발생해 최고치를 기록했다. 영국의 경우 세계에서 3번째로 CCTV가 많으나 전체 뺑소니 사고의 90% 이상이 미해결 사건으로 분류되고 있다. 이는 증거 부족과 인력 부족의 문제로 야기된다[2].

본 논문에서는 이러한 차량 사고에 대한 정보 수집을 도와줄 수 있는 차량 사고 감지 및 관리 시스템을 제안한다. 논문의 구성은 다음과 같다. 2장에서 차량 사고 감지 및 관리 시스템의 설계를 제안하고 3장에서 제안하는 시스템의 구현과 실험 결과를 제시한다. 마지막 4장에서 논문의 결론을 맺고 향후 발전 방향에 대해 살펴본다.

II. 차량 사고 감지 및 관리 시스템의 설계

본 논문에서 제안하는 차량의 사고를 감지하고 관련 데이터를 수집, 가공 저장하기 위한 시스템의 구성도는 그림 1과 같다.



[그림 1] 시스템 구성도

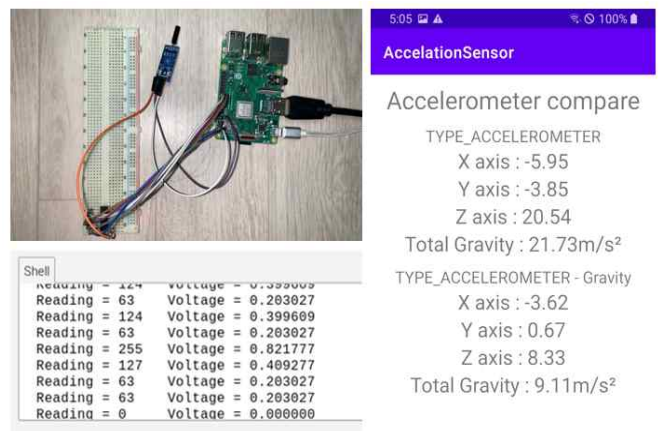
센서부에서는 차량의 사고를 감지하도록 한다. 본 시스템에서는 사고 감지를 위해 충격 센서와 가속도 센서를 활용하도록 한다. 충격 센서는 라즈베리파이 모듈에 연결하여 데이터를 수집하고, 수집한 데이터를 모바일 기기로 전송하도록 하며 가속도 센서는 모바일 기기에 탑재되어 있는 것을 활용하여 데이터를 수집, 가공하도록 한다. 모바일 기기는 사고와 관련한 다양한 데이터를 수집하고 가공하기 위한 용도로 활용되며 본 시스템에서는 스마트폰을 활용하도록 한다. 스마트폰에 탑재된 카메라를 활용하

여 가해 차량의 번호판 이미지를 수집하고 내부 처리 과정을 거쳐 텍스트를 추출하도록 한다. 또한 스마트폰의 위치 인식 기능을 활용하여 사고 위치를 추출하도록 한다. 모바일 기기에서 수집한 사고 관련 정보는 클라우드 저장소에 저장하여 관리되도록 하며 본 시스템에서는 Firebase Storage를 활용하도록 한다.

III. 구현 및 실험 결과

3.1 사고 감지를 위한 센서부

차량 사고를 인식하는 위해 사고 인식 센서가 필요하다. 본 논문에서는 이를 위해 라즈베리파이에 충격 센서를 연동하고 스마트폰의 가속도 센서를 활용하였다. Mobile Device Sensor를 활용하였다. 충격 센서의 경우 충격 발생 시 아날로그 신호를 발생시키고 그 신호의 세기에 따라 일정 값으로 출력된다. 이 때 측정된 값이 설정한 값 이상이면 사고로 판단하도록 하였다. 스마트폰에는 가속도 센서가 탑재되어있어 이를 활용하여 3차원 공간상의 변화를 측정하도록 하여 이를 수치화한 결과를 확인할 수 있도록 하였다.

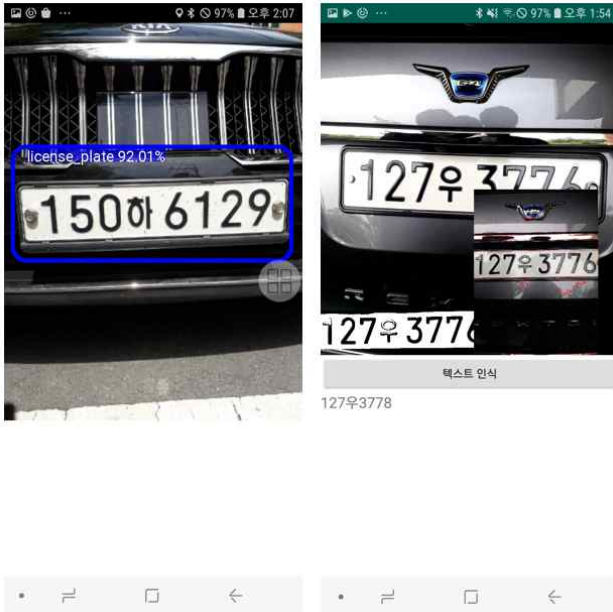


[그림 2] 센서부의 구현 및 실험 결과

그림 2는 센서부의 구현 및 실험 결과를 나타낸 것이다.

3.2 차량 번호 인식을 위한 객체 탐지 및 텍스트 추출

가해 차량의 정보를 수집, 관리하기 위하여 차량의 번호판을 탐지하여 텍스트로 추출하여 저장하도록 하였다. 이를 위해 먼저 차량 번호판 탐지 모듈을 구현하였다. 번호판 탐지를 위해 TensorflowLite를 활용하였으며 학습데이터 구축을 위해 1,500장의 번호판 이미지를 직접 라벨링하였다 [3,4]. Tensorflow는 기계학습을 위한 오픈소스 라이브러리로 제안하는 시스템은 모바일 앱에서 구동시키므로 모바일 버전인 TensorflowLite를 활용하였다.



[그림 3] 객체 탐지 및 텍스트 추출 구현 결과

객체 탐지를 통해 인식된 차량 번호판에서 해당 텍스트를 추출하기 위해 Tesseract를 사용하였다. 이미지로부터 텍스트를 인식하고, 추출하는 기능을 OCR(Optical Character Recognition)이라고 한다. Tesseract는 오프라인 문자인식 기법으로 입력된 이미지의 특징점을 추출하고 그 특징점을 사용하여 이미지에서 텍스트를 인식하고 추출하는 오픈소스 OCR 엔진이다. 그림 3은 객체 탐지 및 텍스트 추출에 대한 구현 및 실험 결과를 나타낸 것이다.

3.3 클라우드 저장소

센서에서 받은 값으로 사고를 감지 후 차량 번호판 객체를 탐지한 이미지와, 차량 번호판 텍스트 추출이 이루어지고 나면 이후 사고에 대한 정보를 저장할 수 있도록 클라우드 저장소에 이미지 파일을 저장하도록 하였다.



[그림 4] Firebase 저장 결과

제안하는 시스템에서는 Firebase Storage를 활용하여 구현하였다. 파일 저장 디렉토리명을 차량 사고 일시로 표기하여 확인할 수 있도록 하였다.

그림 4는 클라우드 저장소에 업로드 된 이미지 파일과 디렉토리명을 나타낸 것이다.

3.4 차량 사고 감지 및 관리 실험 결과

제안하는 시스템의 세부 모듈을 구현하고 통합하여 프로토타입 시스템을 구현하고 실험을 수행하였다. 라즈베리파이 센서부를 통해 차량 사고를 인식하고 스마트폰에서 구현한 앱에서 가해 차량 데이터를 수집, 가공하여 처리하고 클라우드 저장소에 저장하도록 하였다. 구현한 앱의 화면 구성은 다음과 같다. 최상단부터 차례로 사고가 일어난 일자와 시각을 출력하고 Tesseract가 추출한 텍스트 및 인식한 차량 번호판 이미지를 이어서 보여준다. 마지막으로 사고 지점의 위치를 구글맵을 통해 확인할 수 있도록 UI를 구성하였다.



[그림 5] 차량 사고 감지 및 관리 시스템 실험

그림 5는 통합 시스템의 구현 결과의 실험을 수행하는 모습을 나타낸 것이다.

IV. 결 론

최근에도 미해결 뺑소니 사고가 꾸준히 발생하고 있다. 본 논문에서는 차량의 사고를 감지하고 관련 정보를 수집, 저장, 관리하여 사후 처리에 대한 지원이 가능한 시스템을 제안하였다. 제안하는 시스템은 라즈베리파이 연동 충격 센서와 스마트폰의 내부 가속도 센서를 이용하고 차량 사고를 감지하고, TensorflowLite, SSD MoblieNet을 활용하여 자동차 번호판 탐지 모듈을 학습하고 구축하였으며 Tesseract를 통해 번호판 이미지에서 차량 번호 텍스트를 추출하도록 하였다. 또한 수집한 정보 저장을 위해 Firebase Storage를 활용하였다.

향후 과제로 객체 탐지와 번호판 텍스트 추출이 한 단계에 진행이 가능하도록 개선할 예정이며 번호판 추출의 인식률을 높이기 위해 개선된 알고리즘을 적용하고자 한다. 또한 다양한 센서를 적용하여 차량 사고에 대한 세부적인 감지가 가능하도록 개선하고자 한다.

참 고 문 헌

- [1] <https://www.yna.co.kr/view/AKR20180207152800797?input=1195m> (뺑소니 사고 현황 표)
- [2] <https://www.psychreg.org/percentage-hit-and-run-solved/> (영국 뺑소니사고 관련 기사)
- [3] https://github.com/developer0hye/Yolo_Label
- [4] <https://aihub.or.kr/aidata/27727> (차량 번호판 데이터 셋)