

OpenCV 를 활용한 Traffic Light Detection 과 데이터 취합을 통한 운전 습관 알림 시스템

Driving Habit Notification System with Traffic Light Detection using OpenCV

요 약

본 논문은 이미지를 통해서 신호등의 색깔과 정지선을 판별하고 GPS 를 통해서 자동차의 운행정보를 서버에 저장 및 분석한다. 그리고 이를 통해 신호 대기 시 주행 알림을 출력하고 사용자의 운전습관을 알려준다. Raspberry pi 에 연결된 GPS 모듈로 수집된 데이터는 python 을 통해 실시간으로 속도와 가속도에 대한 정보를 생성하고, 카메라 모듈로 수집된 영상은 OpenCV 를 이용해서 이미지를 상하로 배분하여 상부에서는 신호등의 인식, 하부에서는 정지선에 대한 인식을 수행하여 인식 결과를 생성한다. 본 논문을 기반으로 개발된 서비스는 단기적으로는 운전자에게 신호 대기 시의 편리함을 제공하고 장기적으로는 운전자의 운전 습관에 대한 분석을 제공한다.

1. 서 론

스마트폰이 점점 보편화되고 기능이 발달됨에 따라 운전중 신호등 빨간불에 정지시 문자메시지, SNS 사용, 인터넷 서핑, 동영상 시청 등 스마트폰 사용이 증가하고 있다. 그 결과 신호등의 색이 변경된 것을 인지하기 위해 주기적으로 고개를 들어 신호등을 일일이 확인해야하고 신호가 변경된 것을 인지하지 못한 경우 뒷차들에게 크락션이라는 항의를 받게 될 것이다. 이는 사소하게나마 상호 운전자간의 기분을 상하게 할 수 있고, 도로구조가 복잡하거나 러시아워와 같이 차량 운행이 많은 곳에서 원활한 교통흐름에 방해가 된다. 그리고 몇몇 차량의 이러한 사소한 방해가 누적된다면 교통체증의 한 원인이 될 수 있다.

따라서 이를 해결하기 위해 운전자에게 신호의 변화를 알려주고자 한다. 간단히 말하면, 운전선 앞쪽에 설치된 카메라 모듈을 통해서 받아온 이미지가 서버로 입력 되고 서버에서 신호등의 색이 초록색이 된 것을 인식하게 되면 보드에 연결된 버저나 스피커(사운드 출력 장치)를 통해 운전자에게 알림을 주는 것이다.

카메라 모듈로는 영상 데이터를 수집하고 GPS 센서를 통해서 자동차의 위치 데이터를 실시간으로 서버로 전달하여 자동차가 운행중인지 정지중인지 파악한다.

그리고 운행이 정지 되게 된 이후에 서버로 전달된 영상을 분석하여 신호등의 색깔과 정지선 등을 판단한다. 여기서 신호등 색과 정지선 판별은 OpenCV 를 이용하며 신호등 색을 판별한 이후 RED, YELLOW, GREEN 세가지 색 중에서 어느 색에 해당하는 지를 데이터로 저장할 것이다

또한 이렇게 수집한 데이터를 다른 기능에도 활용하는 방안을 생각했다. 위 기술로 수집된 데이터를 이용한다면 운전자가 신호를 어긴 횟수, 시간 등을 파악할 수 있어 운전자의 운전 습관 또한 분석이 가능하다. 이를 위해서 한차례 운행이 종료되면 수집한 데이터 중 검출된 신호는 RED 에 해당되지만 속도가 일정 정수를 초과하는 TIMESTAMP 구간을 합쳐서 해당 운행 동안 총 어느정도 신호를 어겼는지를 시간에 따른 테이블 형태로 출력할 수도 있다. 여기에 추가로 기능을 구현한다면 운전자의 습관을 분석하고 개선하는 방안을 제시할 수 있을 것이다.

하지만 신호등 객체는 영상에서 차지하는 정보량이 적은 편이며, 특히 램프는 더 작기 때문에 검출이 쉽지 않다. 또한 신호등 램프와 유사한 패턴의 객체-도로의 표지판-를 오인식할 소지도 있다. 그러므로 알고리즘의

개발 시 작은 신호등 램프만을 정확히 검출하고 실제 신호등인지를 검증할 수 있는 방안을 고려해야 한다[1].

2. 생성 방법

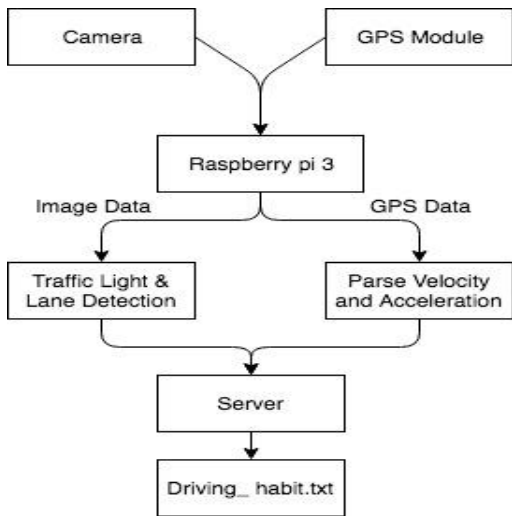


그림 1 전체 시스템 흐름도

본 논문에서 제시하는 시스템의 전체 구조는 [그림 1]과 같다.

2.1 신호등 검출

Camera module 로 읽어 들인 영상을 openCV 를 이용해 신호등과 차선을 검출한다. 신호등은 모양을 적절히 인식하도록 이미지 전처리 과정을 거친다. 즉 컴퓨터가 이미지를 인식하기 쉽도록 바이너리 이미지를 생성 후 신호등 윤곽선 추출 작업을 한다. 실제 촬영된 영상에서는 보통 원색에 가까운 색상은 다소 순화되어 표현되기 때문에 본 기준 색상 값이 이진화 영상에 대한 노이즈를 더 큰 폭으로 줄이는 역할을 할 수 있게 된다[2].

신호등 윤곽선 추출은 cv2.findContours() 함수를 이용한다. 여기서 Contour Hierachy 를 구성할 필요가 없으므로 cv2.RETR_LIST 형식으로 contour line 을 찾는다. 그리고 검출한 신호등 속의 색을 알아내 각 RED, GREEN, BLUE 값이 특정 값 이상일 때 실제 신호등 색인 것을 정의한다. YELLOW 는 RED 와 GREEN 을 조합한다. 하나의 영상 속에서 여러 신호등이 검출될 수 있기 때문에 가까이 있는 신호등을 우선적으로 처리하도록 한다.

```

1 Way to find image contours and color
2
3 //convert to gray image
4 Mat gray_image;
5 cvtColor(image, gray_image, COLOR_BGR2GRAY);
6
7 //convert to binary image
8 Mat bin_image;
9 threshold(gray_image, bin_image);
10
11 Mat canny_image;
12 blur(gray_image, canny_image, size);
13 Canny(canny_image, canny_image, size);
14
15 //detect circle
16 vector<Vec3f> circle;
17 HoughCircles(canny_image, circle, CV_HOUGH_GRADIENT, canny_image.rows);
18
19 // find color of traffic light
20 Mat circle_image = m_image.cone();
21 for(i=0; i < circle.size(); i++){
22     Point center(cvRound(circles[i][0]),cvRound(circles[i][1]));
23     int radius = cvRound(circles[i][2]);
24
25     int k=circles[i][0]; //x
26     int j=circles[i][1]; //y
27
28     int BLUE=Img.at<Vec3b>(j,k)[0];
29     int GREEN=Img.at<Vec3b>(j,k)[1];
30     int RED=Img.at<Vec3b>(j,k)[2];
31 }
  
```

그림 2 신호등에서 원과 색을 추출하는 코드

또한 야간에는 반대편의 전조등이나 앞차의 후미등 불빛으로 오인이 일어날 수도 있는데 이를 위해 야간차량 검출이 필요하다. 이를 위해 2 가지 방법을 사용할 수 있는데 일반적으로 자주 사용되는 영상, 즉 자동 노출(autoexposure)설정으로 촬영되는 영상에서 밝은 영역을 검출한 후 해당 영역이 실제 전조등 또는 후미등인지를 appearance 기반으로 확인하는 것과 다른 저노출(lowexposure) 영상에서의 thresholding 을 통해 강한 밝기값을 갖는 light blob 만을 검출하는 방법이다[3].

2.2 차선 검출

실제 운전 상황에서는 신호등이 안보이는 상황이 존재할 뿐더러 차선을 보고 교통상황을 아는 경우도 있다. 그렇기 때문에 도로 위의 선을 검출해 교통상황 인식에 이용한다면 정확성을 높일 수 있다.

차선, 정지선 등은 openCV 에서 Hough Transformation 알고리즘을 이용해 검출할 수 있다. Hough Transformation 은 픽셀 값이 비슷한 것과 같이 특성이 비슷한 픽셀에 대해 각 점을 지나는 직선의 값 중 가장 많이 counting 된 이 이미지에서 직선일 확률이 높다는 원리이다. 여기서 실시간으로 영상을 분석해야 하는 본 기술에서 모든 이미지에 대해 직선을 검출하는 것은 상대적으로 비효율적이다. 그렇기 때문에 무작위로 픽셀을 선택해 변환함으로써 효율성도 갖추고 직선도 충분히 도출하는 방식인 Probabilistic Hough Transformation 을 사용한다. 이는 cv2.HoughLinesP() 함수에 내장되어 있다.

이렇게 검출된 차선을 이용해 현재 있는 차선과 다른 차량에 가려져 검출되지 않은 차선까지 검출함으로써 신호등 검출에서 알기 어려운 주행 신호를 알 수 있다. 그리고 이 기술을 Driving_habit 생성에도 이용할 수 있을 것이다. 즉 신호등 검출에서 알 수 없는 신호까지 알게 되어 정확성을 높일 수 있다.

```
lines=cv2.HoughLinesP(edges, 1, np.pi/180,
                        threshold, minLength, maxLength)
```

그림 3 차선검출에 사용하는 HoughLinesP ()형태

등 제어용 야간 차량 검출 개선”, 한국자동차공학회 학술대회 및 전시회, 3권, 11호, pp.1269-1274, 2011

2.3 Traffic light detection

2.1, 2.2 를 통해 영상에서 검출된 신호와 더불어 차선을 안다면 차량이 주행을 해도 되는지에 대한 여부를 알 수 있다. 이를 GPS data 와 결합하여

```
Car_velocity==0 && Traffic_Light=='GREEN'(주행 가능)
```

일 경우 알림을 출력한다. 이는 raspberry pi 에 연결된 사운드 출력 장치에 신호를 보냄으로써 구현할 수 있다.

2.4 Driving habit 분석

운전습관 분석 시스템은 차량 운행이 종료된 후 운행 중에 분석된 Car_velocity, Traffic_Light 를 이용한다. 우선적으로는 신호 위반을 조사하는데, 동일한 TIMESTAMP 구간에서

```
Car_velocity!=0 && Traffic_Light=='RED'(주행 불가)
```

인 빈도를 조사해 주행 종료 후 textfile 에 기록한다.

3. 결론 및 향후 계획

본 논문에서는 오픈소스 라이브러리인 OpenCV 기반으로 신호등의 정보와 차선의 정보를 추출하여 사용자에게 실시간으로 알림을 주고 운행이 종료된 후에 신호등을 위반하여 운행한 경우를 데이터로 정리한 Driving_habit.txt 를 생성하는 방법을 제시하였다. 본 논문에서는 Driving habit 으로 신호 위반만을 제시하였지만 GPS Module 을 통해서 Acceleration 정보도 수집이 가능하기 때문에 이 데이터를 장기적으로 수집하여 경향성을 파악할 예정이다. 이를 통해서 양이나 음의 값으로 크기가 큰 경향성을 벗어나는 데이터를 통해 급가속과 급정거의 경우를 파악하여 이를 Driving habit 으로도 제공할 계획이다.

감사의 글

“본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음”(2015-0-00910)

참고문헌

- [1] 김용권, 이기성, 조성익, 박정호, 최경호, “차세대 실감 내비게이션을 위한 실시간 신호등 및 표지판 객체 인식”, 한국공간정보시스템학회논문지, 제10권, 제2호, pp.13-24, 2008.
- [2] 고승현, 색상 검출 및 딥러닝을 이용한 실시간 신호등 인식방법, 연세대학교 대학원, 제1권, 제1호, pp.10-22, 2017
- [3] 음성민, 정호기, “차선 인식을 활용한 지능형 전조