

# 어린이 보호구역 교통사고 방지 시스템

최지훈, 임현창, 정재천, 공기석  
한국산업기술대학교 컴퓨터공학과  
e-mail:{rw607, dlafkaqh, jjc100, kskong}@kpu.ac.kr

## Prevention System of Car Accident in Children Protection Zone

JiHun Choi\*, HyunChang Im, JaeCheon Jung, Ki-Sok Kong  
Dept. of Computer Engineering, Korea Polytechnic University

### 요 약

교통사고를 예방하고자 정부에서는 지속적인 정책을 펼치고 있으나 발생률이 크게 줄지 않는 실정이다. 본 논문에서는 이러한 한계점을 보완하고자 영상처리를 이용하여 교통사고를 줄일 방법을 제안하고자 한다. 우선 어린이 보호 구역 내에 설치된 카메라를 이용하여 교통사고 발생 위험을 판단하고, 보행자 및 차량 운전자에게 알려 줌으로써 교통사고 발생을 감소하고자 하는 시스템을 설계하였다. 또한 실시간 영상처리를 위해 관심 영역 지정, 멀티 스레드 등 다양한 방법을 적용해 보았으며, 위험 상황 탐지 시 탐지 결과를 주변 운전자에게 전송하려는 방법을 제시하고 실제 구현하여 확인해 보았다.

### 1. 서 론

어린이의 교통안전을 위해 1995년부터 교통안전시설 및 도로부속시설을 설치하여 자동차의 통행속도를 제한하는 어린이보호구역이 만들어졌다[1].

그러나 2013년 도로교통공단이 공개한 자료에 따르면 안전을타리 설치, 과속카메라를 이용한 속도 알림 시스템, 과속방지턱 등 교통사고를 예방하는 많은 방법이 도입되었음에도 어린이 보호구역 내 교통사고 발생률은 크게 줄어들지 않는 실정이다.

특히 어린이 보호구역 내 교통사고의 주된 원인은 '운전자의 안전운전 불이행(43.7%)'이며, 사고 유형 중 '횡단 중 사고(67%)'의 경우가 가장 많이 발생했다[2].

이를 보완하기 위해 보행자 자동인식 신호기, 자동 긴급 제동 시스템(AEB: Autonomous Emergency Braking) 등 여러 가지 시스템이 운영되었으나 설치 위치 등에 한계점이 있는 경우가 많았다.

따라서 본 연구는 어린이 보호구역의 카메라를 이용하여 보행자가 도로를 횡단할 때 운전자에게 실시간으로 상황을 알려주는 방법을 구현하고자 한다.

이를 위해 어린이 보호구역 도로상에 보행자가 있는 경우와 보행자가 도로로 갑작스럽게 접근하는 경우를 CCTV(Closed Circuit Television)로 촬영하여 위험 상황을 판별한다. 판별된 데이터는 도로상에 설치된 전광판을 이용하여 운전자가 인지하게 한다. 또한, 위험 상황이 발생한 보호구역 근처 200m 반경에 있는 운전자의 모바일 애플리케이션으로 통보하여 주의를 주는 시스템을 구

현함으로써 교통사고 발생률을 줄이는 데 연구의 목적이 있다.

### 2. 관련 사례

현재 실시간으로 상황을 알려주어 교통사고를 방지하는 시스템으로는 보행자 자동인식 신호기와 자동긴급제동시스템이 있다.

보행자 자동인식 신호기는 보행자가 횡단보도 대기 공간에서 보행자 작동 신호기의 버튼을 작동하지 않아도 자동으로 차도 횡단을 목적으로 하는 보행자를 감지하여 보행 신호를 요청하는 장치이다[3]. 그러나 보행자 자동인식 신호기는 횡단보도 대기 공간의 보행자가 횡단을 목적으로 하는지 알 수 없고, 차량을 고려하지 않아 교통체증을 유발할 수 있다.

자동긴급제동시스템은 차량에 카메라나 센서를 부착하여 전방의 물체나 보행자를 인식해 능동적으로 정지하거나 감속하여 피해를 줄이는 안전 시스템이다. 그러나 모든 차량에 탑재하여야 하므로 사회 전반적으로 큰 비용을 초래하고, 차량이 아닌 보행자나 자전거, 오토바이 등은 잘 인식하지 못한다는 한계가 존재한다. 또한, 전방 시야가 확보되지 않거나 굽은 길에서는 오작동 가능성이 있으며, 오작동 시 더 큰 사고를 발생시킬 수 있다[4].

따라서 본 연구는 이러한 점을 보완하고자 도로의 CCTV를 이용하여 영상처리 기반의 교통사고 방지 시스템을 구현하고자 한다. 이를 구현하기 위해 영상처리는 OpenCV를 사용하였고, 운전자의 위험 상황 인지를 위해 안드로이드 기반의 애플리케이션을 제작하였다[5].

### 3. 세부 설계 및 구현

#### 3.1 시스템 구성



(그림 1) 시스템 동작

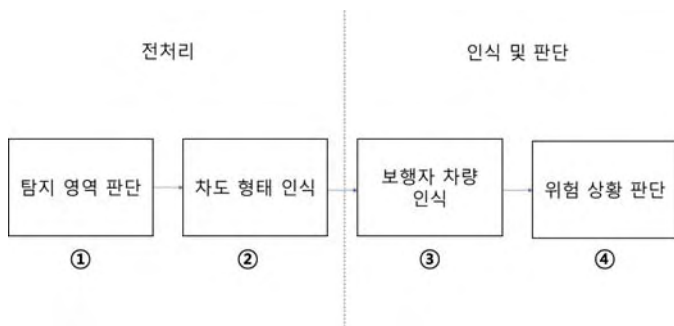
본 연구에서 구현한 시스템은 (그림 1)과 같이 영상을 촬영하고 위험 상황을 판단하는 보드와 이를 관리하는 중앙 서버, 운전자에게 상황을 알리고 편의 기능을 제공하는 모바일 애플리케이션으로 크게 총 3개의 요소로 구성된다.

도로 위의 보드는 연결된 카메라를 이용하여 도로를 촬영한다. 이때 도로 위의 보행자와 차량을 탐지하고 그 결과로 위험 상황을 판단한다. 판단한 결과는 보드에 연결된 LCD 표지판에 표시하며, 중앙 서버로도 데이터를 전송한다.

서버에서는 수신한 데이터를 이용하여 위험 상황이 발생한 위치를 파악하고 운전자 모바일 애플리케이션으로 데이터를 전달한다.

운전자의 모바일 애플리케이션은 알람 표시 여부와 알람 방식을 결정하고, 운전자가 위험 상황을 인지하도록 한다.

#### 3.2 데이터 가공 및 분석 방법



(그림 2) 영상처리 동작

도로에 설치된 보드는 (그림 2)에서와 같이 연결된 카메라에서 탐지 영역과 차도 형태를 인식하기 위한 목적의 전처리 과정을 거친다. 탐지 영역은 보행자와 차량이 모두 탐지될 수 있는 영역을 의미하고, 차도 형태는 차

량이 탐지될 수 있는 영역을 의미한다.

(그림 2)의 ①로 표현된 탐지 영역의 파악은 Background Subtractor를 이용한다[6]. 이 작업에서는 1초간 수집된 프레임을 이용하여 움직임이 발생한 프레임을 만든다. 프레임을 만드는 작업은 지정된 시간 동안 반복되며, 만들어진 프레임들은 OR 연산으로 누적하여 저장한다. 최종 저장 시에는 Opening 연산(침식 후 팽창 연산)을 수행하여 탐지 영역을 명확하게 한다.

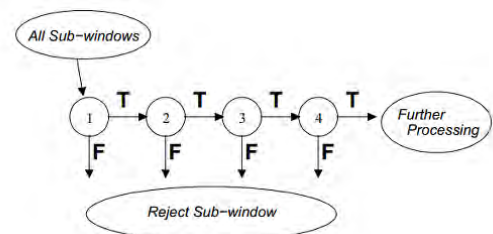
(그림 2)의 ②로 표현된 차도 형태의 판단은 탐지 영역의 파악이 종료된 후 수행한다. 이 과정에서는 차량을 인식하기 시작하며, 차도의 형태를 저장하기 위한 이미지를 생성한다. 이 이미지에는 지정된 시간 동안 인식된 차량의 위치에 1px 크기의 선을 그리고, 차량의 움직임에 따라 누적한다. 지정된 시간에 도달하면 선을 누적한 이미지에 침식 연산과 팽창 연산을 수행한다. 이로써 탐지 오류로 인한 잡음을 최소화하고, 평균값을 도출하여 실제 차도의 형태에 근사하게 하였다.

#### 3.3 데이터 분석 과정

보드에서 탐지 영역과 차도 형태를 파악하는 전처리 과정이 종료되면 (그림 2)의 ③처럼 실시간으로 촬영한 영상을 이용하여 보행자와 차량을 인식한다. 이때 탐지 누락이 존재하면 심각한 결함이 될 수 있으므로, 탐지 오류율보다는 탐지 누락을 줄이는 것에 더 초점을 맞추어 실시간 영상에서 보행자와 차량을 인식하도록 구현하였다.

이 과정에서는 실시간으로 촬영한 원본 영상과 전처리 과정에서 생성한 탐지 영역을 AND 연산한다. 이 결과 보행자와 차량이 탐지될 수 있는 부분을 지정하는 관심 영역이 지정(ROI : Range of interest)된다.

ROI 된 영상은 흑백 영상으로 변경하고, 히스토그램 평준화를 수행한다. 이 과정에서 영상의 빛에 의한 변형을 줄이고, 개체의 경계선을 뚜렷하게 만든다. 이후 보행자와 차량을 검출하는 각각의 검출기를 이용해 별도 스퀘드에서 검출한다.



(그림 3) Cascade Classifier[7]

검출 방식은 (그림 3)의 Cascade Classifier를 사용하였다. 이 방식은 여러 개의 검출기를 연결해서 순차 탐색하는 방법이다. 처음에는 간단한 검출기를 적용하여

개체를 추출하고 점점 더 복잡한 검출기를 적용하여 정확성을 높인다. 그럼으로써 속도와 검출율에서 단일 검출기를 사용하는 방법보다 우수한 성능을 낼 수 있도록 했다. Cascade Classifier에 사용된 각 검출기로는 영상의 형태를 이용해 개체를 탐지하는 LBP(Local Binary Pattern detection)를 사용하였다[8].

```
// Exclude areas excluding road areas in the original image.
bgMask.locateForeground(img, fgimg);

// Detect pedestrians and vehicle
std::thread t1(detectObjects, std::ref(pe_Detector), std::ref(fgimg));
std::thread t2(detectObjects, std::ref(car_Detector), std::ref(fgimg));
t1.join();
t2.join();

// Judge the situation of the road
situation.sendPredictedSituation( pe_Detector.getFoundObjects(), car_Detector.isFound() );
```

(그림 4) 보행자 및 차량 탐지 과정

(그림 4)는 위 과정들을 구현한 부분이다. locateForeground 메소드는 보행자와 차량을 원활하게 탐지하기 위해 영상을 ROI하고 흑백 영상으로 전환 후, 히스토그램 평준화를 수행한다. 이후 별도의 스레드에서 detectObjects 메소드를 이용해 보행자와 차량을 각각 탐지하였다.

### 3.4 구현 결과



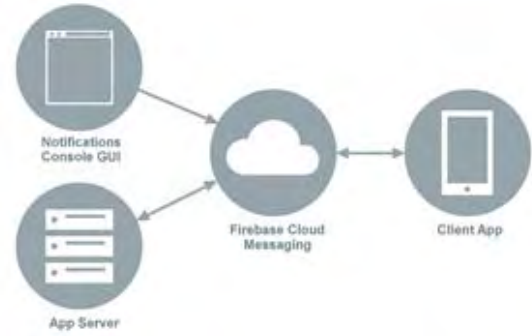
(그림 5) 보행자 및 차량 인식 결과

(그림 2)의 ④에 해당하는 위험 상황 판단은 인식 과정을 통해 (그림 5)처럼 획득한 보행자와 차량의 위치를 기준으로 판단하였다.

판단의 기준은 인식 과정에서 검출되는 보행자의 평균 너비 값이 실제 측정 수치로 75cm인 점을 고려하여 판단하였다. 보행자가 차도 영역의 약 150cm 이내로 접근할 시 주의(caution)상황으로 판단하였다. 또 보행자로 인식된 객체의 가운데와 가운데 바닥 사이 5개 점 중 4개 점 이상이 차도 영역 내인 것으로 판단될 시 경고(dangerous)상황으로 판단하였다.

이런 기준으로 위험 상황이 발생했다고 판단될 시 도로에 설치된 LCD 표지판으로 경고하며, 주변 차량의 존재 여부에 따라 중앙 서버를 통해 주위 운전자의 모바일로 위험 상황을 전달한다.

이 과정은 (그림 4)의 sendPredictedSituation 메소드에서 수행한다.



(그림 6) Firebase Cloud Message[9]

위험 상황 전달은 위험 상황을 인지한 도로 위 보드에서 중앙 서버로 데이터를 전달한다. 전달하는 데이터는 위험 상황과 각 보드에 할당된 ID 값이다. 서버는 수신한 ID 값을 이용해 카메라 설치 데이터베이스에서 카메라가 설치된 지역의 위도, 경도 값을 알아낸다. 그리고 알아낸 위치 정보를 포함하여 (그림 6)과 같이 Firebase Cloud Server에 FCM(Firebase Cloud Message) 전송을 요청한다.

운전자 모바일은 상시 백그라운드 작업인 서비스 컴포넌트에서 FCM 데이터를 수신한다. 수신한 데이터에서 교통 위험 상황이 발생한 위치가 모바일 기기 주변 200m 이내인 경우 운전자(모바일 기기 사용자)에게 경고를 보여준다.



(그림 7) 운전자 모바일 화면

운전자의 모바일 애플리케이션은 운전자의 편의를 고려하여 기본적인 차량용 네비게이션 기능을 탑재하였다. 운전자가 네비게이션 기능을 사용하는 중에 200m 이내에서 교통 위험 상황이 감지된 경우, 네비게이션 화면에서 음성 경고와 함께 위험 상황이 발생한 위치를 (그림 7)과 같이 지도상에 표시한다.

또한, 운전자가 네비게이션 기능을 사용하지 않는 경우에도 Notification 기능을 이용하여 운전자에게 경고할 수 있도록 했다. 이때 경고 방식은 운전 중에도 무리 없

이 확인할 수 있도록 음성과 메시지로 구현했다.

**참고문헌**

<표 1> 위험 상황 전송 시간 측정 결과

	검출 시간	판단 시간	전송 시간
평균 시간(ms)	115	7	792

<표 1>은 위험 상황이 발생한 후 운전자의 모바일로 전달되기까지 걸리는 시간을 100번의 테스트를 통해 평균을 도출한 결과이다. 보행자 및 차량을 검출하는데 걸리는 시간은 약 115ms이고, 검출한 보행자와 차량을 이용해서 위험 상황을 판단하기까지 걸리는 시간은 7ms로 측정되었다. 그리고 운전자의 모바일로 경고 메시지가 도달하기까지 걸리는 시간은 792ms로 측정되었다. 즉, 본 시스템은 위험 상황이 발생한 후 총 약 0.91초 이내에 주위의 운전자에게 주의를 줄 수 있다.

**4. 결론 및 향후 과제**

본 연구는 영상처리를 이용한 어린이 보호구역 교통 사고 방지 시스템을 구현하였으며, 실시간으로 운전자에게 위험 사항을 전달할 수 있는 프로세스를 구축하였다.

본 연구의 의의는 다음과 같다. 첫째, 교통사고에 대한 책임을 민간단체와 학교에 부여하던 것을 해결하고 인력이 있어야 하는 불편을 본 연구를 통해 해소할 수 있다.

둘째, 교통 방지 해결 방안 중 보행자 자동인식 신호기나 자동 긴급 제동 시스템과는 달리 일반적인 어린이 보호구역에 바로 설치할 수 있으며, 비교적 저렴한 비용으로 어린이의 교통안전을 확보하고, 차량 외부의 카메라를 이용하여 사각지대를 줄일 수 있다.

셋째, 향후에 쌓인 데이터를 통해 교통사고 위험 사항을 분석하여 위험지역을 파악하고 예방할 수 있다.

본 연구에서 구현한 시스템은 실시간으로 데이터를 전송하는데 초점을 두었다. 그래서 보행자와 차량을 인식할 때 검출기의 데이터를 갱신하거나 탐지 영역과 차도 형태를 파악하는 동작을 지속하지 않는다. 그러므로 차량의 형태가 다양해지거나 차도의 형태가 바뀌거나 혹은 카메라의 촬영 각도가 변경될 시 기존 정보를 갱신할 필요가 있다.

또한, 실제 도로에서 오랜 시간 촬영하는데 제한이 있어 교통사고의 감소율이나 탐지의 신뢰성을 측정하는데 어려움을 겪었다. 향후 연구에서는 이를 보완하기 위해 좀 더 세밀한 보행자 및 차량 검출기를 적용하고 실제 도로에 적용해보는 과제가 남아있다.

[1] 도로교통안전관리공단, “어린이 보호구역 개선사업 업무편람”, 2004.1.  
 [2] 김태정, “스쿨존(어린이 보호구역)내 어린이교통사고 특성 분석”, 도로교통공단, 2013.4.13.  
 [3] 경찰청, “보행자 자동인식 신호기 표준지침”, 2013.9.13.  
 [4] 자동 긴급제동시스템(AEB), “<http://www.kasdi.co.kr/drivingcommon/2780>”.  
 [5] OpenCV, “<http://opencv.org/>”.  
 [6] Andrew B Godbehere, Akihiro Matsukawa, Ken Goldberg. “Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation”. American Control Conference, 2012, pp.4305 - 4312.  
 [7] P. Viola and M. J. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, Computer Vision and Pattern Recognition, 2001, vol.1. pp.511-518.  
 [8]Timo Ahonen, Abdenour Hadid, Matti Pietikäinen, “Face Description with Local Binary Patterns: Application to Face Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, vol.28. pp.2037-2041.  
 [9] Firebase Cloud Message, “<https://firebase.google.com/docs/cloud-messaging>”.