

딥러닝을 활용한 영상기반 교통사고 예방 안전시스템

한 승 헌*, 채 석°, 박 재 한*, 사이드 알리 하산*, 타릭 라힘*, 신 수 용*

Video-Based Traffic Accident Prevention Safety System Using Deep Learning

Seung Heon Han*, Seog Chae°, Jae Han Park*, Syed Ali Hassan*,
Tariq Rahim*, Soo Young Shin*

요 약

본 논문은 YOLO(You Only Look Once)와 깊이 카메라를 이용한 차량과 보행자 간 사고방지 시스템을 제안한다. 딥러닝 기반의 YOLOv2-tiny를 이용한 차선과 보행자의 인식정보를 기반으로 보행자가 차도로 들어오는 상황에서 사용자에게 사전에 위기 상황을 알려준다. 기존의 관련 연구에서 사용한 이미지 처리 방식은 알고리즘이 복잡하므로 제안하는 시스템은 딥러닝을 이용하여 시스템 복잡성을 개선하였다. 또한, 딥러닝을 이용하여 실시간으로 물체 인식을 이용해 차선과 보행자를 동시에 인식했다. 그리고 점선을 이용하여 현재 주행 중인 차선을 확인하였다. 보행자와 사용자 간의 거리를 계산하기 위해 깊이 카메라를 사용하였으며 사고 상황 인식 알고리즘을 개발하였다. 본 시스템은 2차선을 기준으로 자동차 주행 시 보행자 충돌 사고를 중점으로 구현하였다. 마지막으로 실험 결과를 통해 제안한 시스템의 인식 정확도를 확인할 수 있으며 향후 연구 방향에 대해 논의한다.

Key Words : Accident prevention system, Deep learning, Depth Camera, Lanes, Pedestrians, YOLOv2-tiny

ABSTRACT

Intelligent visual monitoring for road vehicles and pedestrians is the key to developing autonomous intelligent traffic systems. Lately, traffic incident detection using image processing and computer vision has drawn much attention. In this paper, an accident prevention system based on deep learning (DL) i.e., YOLOv2-tiny is proposed. The DL-based accident prevention system detects lanes and pedestrians simultaneously, followed by a depth camera that estimates the distance between the pedestrian and the user. The current position of the vehicle is decided on the dot lane marker as a reference. Based on the distance calculation, an accident situation recognition algorithm is developed for an accident prevention system. Detailed experimental results for lanes and pedestrian detection are provided, showing the effectiveness of the proposed DL-based accident prevention system.

1. 서 론

현재 차량과 보행자 간 교통사고는 큰 노력에도 불

구하고 꾸준히 발생하며, 매년 발생 건수는 정체되어 교육/학습적인 방법으로는 교통사고를 줄이는 것은 한계가 있다. 또한, 최근 사고 예방 연구도 꾸준히 진행

* 이 연구는 금오공과대학교 학술연구비에 의하여 지원된 논문입니다. (2019-104-122)

• First Author : Kumoh National Institute of Technology, gjs9410@kumoh.ac.kr, 학생(석사), 정회원

° Corresponding Author : First Author : Kumoh National Institute of Technology, schae@kumoh.ac.kr, 정교수, 정회원

* Kumoh National Institute of Technology, qkrwogks7094@kumoh.ac.kr, 학생(석사), 정회원; syedali@kumoh.ac.kr, 학생(석사), 정회원; tariqrahim@kumoh.ac.kr, 학생(박사), 정회원; wdragon@kumoh.ac.kr, 부교수, 종신회원

논문번호 : 202005-099-D-RU, Received May 1, 2020; Revised May 21, 2020; Accepted May 25, 2020

되고 있다. 3D 모델 기반 차량 추적을 사용한 교통사고 예측¹¹, ResNet(Residual Network)을 이용한 도시 교통 사고 예측¹², VANET(Vehicular Adhoc Network)의 컨벌루션 뉴럴 네트워크를 기반으로 한 교통사고 예측¹³ 등이 있다.

차선, 보행자 등을 인식하기 위해서 최근 물체 인식 분야도 활발히 연구되고 있다. 기존 연구로는 영상특징점(Image features)과 라인 피팅모델(Line fitting model)을 사용한 차선 인식¹⁴, 이중 색 공간을 기반으로 한 차선 인식¹⁵ 등이 있다. 이러한 연구들은 이미지 처리 방식이 사용되었으며, 알고리즘이 복잡하고 정확도가 떨어지는 문제가 있다. 이를 해결하기 위해 최근 딥러닝을 이용한 차선 인식 연구가 이루어지고 있다^{16,7}.

본 논문에서는 딥러닝을 활용한 교통사고 예방 안전시스템을 제안한다. 딥러닝을 활용한 객체 탐지 기법 중 YOLO(You only look once)를 이용하여 실시간으로 차선과 보행자를 인식한다. 그리고 사고방지를 위해 깊이 카메라를 이용하여 보행자와의 거리를 측정한다. 인식된 정보를 바탕으로 주행 중인 차선 인식 및 사고 상황 인식 알고리즘을 제안하고, 구현을 통해 실제 2차선 도로 환경에서 성능을 확인하였다.

II. 본 론

2.1 시스템 개요

그림 1과 2는 각각 시스템 구성, 전체 시스템 모델을 나타낸다. 차선 및 보행자 인식을 위해 클래스를 황색 선(Yellow lane), 백색 선(White lane), 점선(Dot lane), 사람(Person) 총 4개로 정한 후, 라벨링(Labeling) 과정을 진행한다. 그리고 YOLO 모델을 통해 차선과 보행자를 인식한다. 인식된 점선(Dot

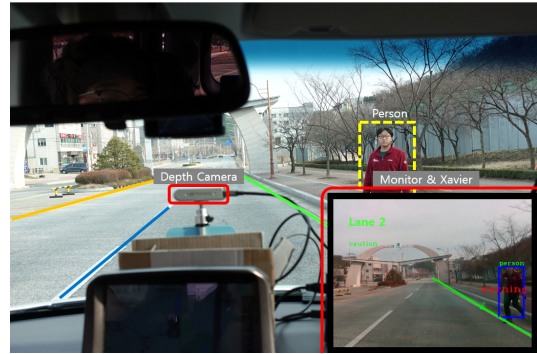


그림 1. 시스템 구성도
Fig. 1. System configuration

lane)을 이용하여 주행 차선을 인식하고, 인식된 보행자를 기준으로 깊이 카메라를 이용한 보행자와의 거리를 측정한다. 그리고 보행자와의 거리와 도로 진입 여부에 따른 사고 상황 인식 알고리즘을 제안한다. 최종적으로 사고 상황 인식 알고리즘을 통해 사고 상황이 예측되면 운전자에게 위험 알람을 제공하는 전체 시스템을 설계 및 구현하였다.

2.2 차선, 보행자 인식 및 네트워크 구조

2.2.1 차선 및 보행자 인식

본 논문에서는 2차선 도로를 기준으로 하였다. 클래스는 황색 선, 백색 선, 점선, 사람 총 4가지로 이루어져 있다. 데이터 세트는 차량에 직접 깊이 카메라를 장착하여 1000장을 수집하였고, 라벨링은 labelImg 도구를 이용하였다. 사고 상황 인식 알고리즘을 위해서 백색 선, 사람이 필요하다. 황색 선, 점선은 몇 차선을 주행 중인지 알기 위해서, 알고리즘 확장을 위해서 필요하다.

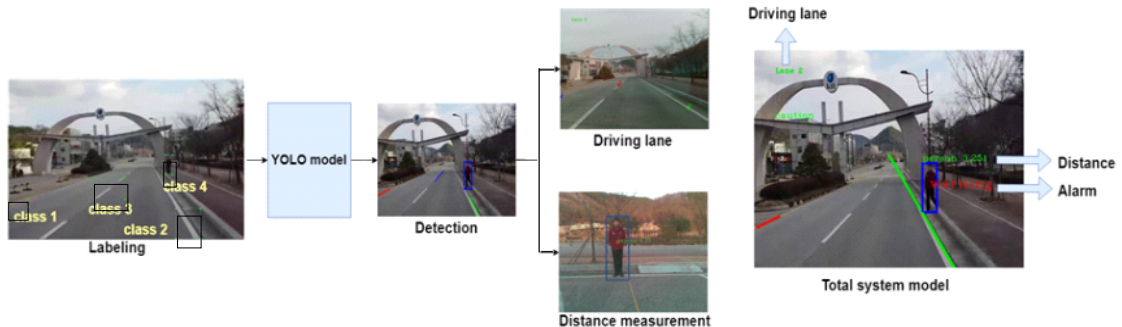


그림 2. 전체 시스템 모델
Fig. 2. Total system model

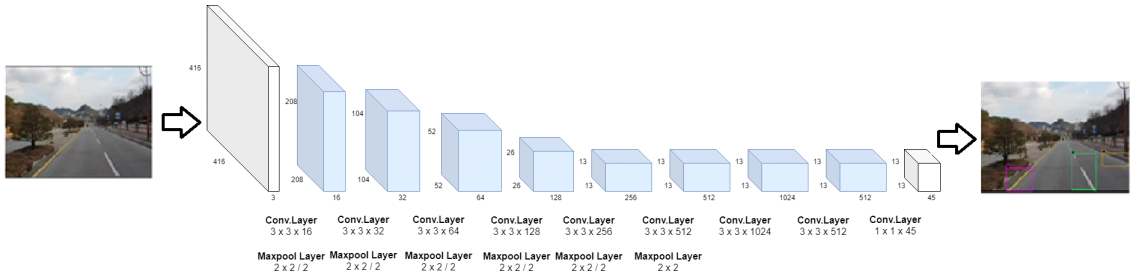


그림 3. YOLOv2-tiny 네트워크 구조
Fig. 3. YOLOv2-tiny network architecture

2.2.2 YOLO 네트워크 구조

차선과 보행자 인식을 위해 YOLOv2-tiny^[8]를 사용하였다. 딥러닝 모델은 실시간 차선 및 보행자 인식을 위해서 빠른 객체 탐지 성능과 객체 탐지 계산이 이루어지는 임베디드 보드인 Xavier과 적합성을 고려하였다.

YOLOv2-tiny의 네트워크 구조는 총 9개의 컨벌루션 계층(convolutional layer)으로 이루어진다. 필터 사이즈(Filter size)는 5*(5+클래스 수) 식에 따라 클래스가 총 4개이므로 45개로 정하였다. 자세한 네트워크 구조는 그림 3에 나타난다.

2.3 주행도로 인식 알고리즘

차량과 보행자 간 사고 상황 인식 알고리즘 개발을 위해서는 몇 번째 차선을 주행 중인지 알 필요가 있다. 주행 중인 차선의 전체 차선 수는 알고 있다고 가정한다. 제안하는 주행도로 인식 알고리즘은 알고리즘 1에 나타난다. 알고리즘은 황색, 또는 백색 선이 인식 되면 시작된다. 인식된 점선(dot_lane)의 수를 n, k번째 점선(dot_lane)의 x 좌표를 x_k로 정의한다. 또한, 중앙점 기준 가장 가까운 왼쪽 점선(dot_lane)의 x 좌표를 x_L, 오른쪽 점선(dot_lane)의 x 좌표를 x_R로 정의한다. 황색 선 인식 시 현재 주행 중인 차선은 x_L이 존재할 경우 2차선, 그렇지 않으면 1차선이 된다. 백색 선 인식 시 현재 주행 중인 차선은 x_R이 존재할 경우 (전체 차선 - 1), 그렇지 않으면 전체 차선으로 판단한다. x_L이 중앙점 기준 오른쪽으로 이동하면 현재 주행 차선을 -1하여 출력한다. x_R이 중앙점 기준 왼쪽으로 이동하면 현재 주행 차선을 +1하여 출력한다.

2.4 거리측정 시스템

그림 4는 깊이 카메라의 출력 결과이다. (a)는 Color Sensor 기반 RGB 이미지, (b)는 깊이 이미지이다. 거리측정은 Intel에서 지원하는 pyrealsense2^[9]를

사용했다. pyrealsense2는 깊이 이미지의 모든 픽셀에 대해 거리가 측정되고, 그림 (b)처럼 측정된 거리에 따라 색이 다르게 출력된다. 거리 계산 방법은 그림 5와 같다. 먼저 일반 카메라 이미지 (a)를 추출하여

알고리즘 1. Driving road recognition algorithm

```

1: Input: yellow_lane, white_lane, dot_lane
2: Output: current_lane
3: Initialization for lane identification:
4:   n = the number of the detected dot_lane
5:   x_k = the x-axis value of k-th dot_lane (1<=k<=n)
6:   x_center = the x-axis value of mid-point in camera FOV
7:   x_L = the x-axis value of the nearest left dot_lane
8:   x_R = the x-axis value of the nearest right dot_lane
9:   current_lane, x_L, x_R = 0
10:  def fine_the_nearest_line()
11:    for (i = 1; i < k+1; i++){
12:      if x_i > x_center*(1/2) and x_i < x_center
13:        x_L = x_i
14:      else if x_i < x_center*(3/2) and x_i >
x_center
15:        x_R = x_i
16:    }
17:  while():
18:    fine_the_nearest_line()
19:    if yellow lane is detected then
20:      if x_L > 0 then
21:        current_lane ← 2
22:      else
23:        current_lane ← 1
24:    if white lane is detected then
25:      if x_R > 0 then
26:        current_lane ← total lane - 1
27:      else
28:        current_lane ← total lane
29:    if the x_L crosses the center then
30:      current_lane ← current_lane - 1
31:    if the x_R crosses the center then
32:      current_lane ← current_lane + 1
33:  end
    
```

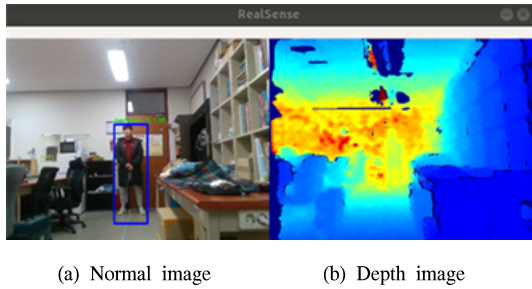


그림 4. RealSense 카메라의 출력 이미지
Fig. 4. Output image of RealSense camera

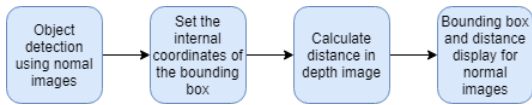


그림 5. 거리측정 및 표기
Fig. 5. Distance measurement and notation

YOLOv2-tiny를 이용해 사람을 인식한다. 바운딩 박스의 좌표가 생성되면 거리측정을 위해 특정 좌표를 선택하고, 깊이 이미지 (b) 상에 대응하는 좌표의 거리를 계산한다. 최종적으로 계산한 거리와 바운딩 박스는 일반 이미지 (a) 상에 표시한다.

일반적으로 거리측정을 위해서 바운딩 박스의 중점을 기준 좌표로 선택하지만, 객체 탐지 목표가 사람이므로 성능 향상을 위해 바운딩 박스의 기준 좌표의 변경이 필요하다^[9]. 인식된 물체와 카메라 사이에 장애물이 있는 경우 즉, 바운딩 박스의 중점과 겹쳐질 때의 거리측정 오류를 해결하기 위해 상체의 좌표를 이용한 거리측정 방식을 이용하였다. 개선 결과는 그림 6에 나타낸다.



그림 6. 장애물이 있는 경우 거리측정 개선
Fig. 6. Improved distance measurement in the presence of obstacles

2.5 사고 상황 인식 알고리즘

일반적으로 보행자 사고는 도보와 가까운 마지막 차선 부근에서 발생하기 때문에 마지막 차선을 기준

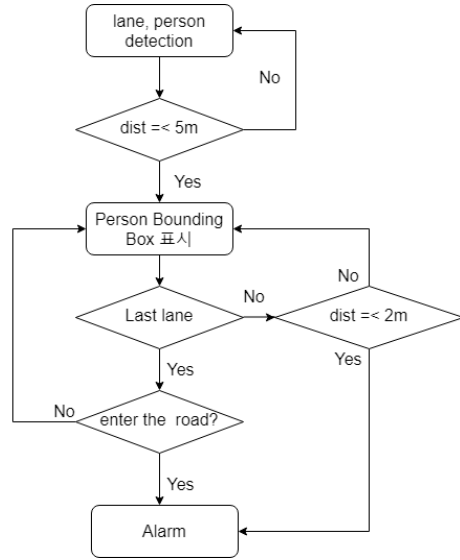


그림 7. 사고 상황 인식 알고리즘
Fig. 7. Accident situation recognition algorithm

으로 알고리즘을 개발했다. 알고리즘은 차량과 보행자 간 거리가 일정 수준 이하로 가까워질 때, 또는 보행자의 차도 진입 여부를 판단하여 실행된다. 알고리즘은 그림 7과 같은 순서로 수행된다. 깊이 카메라에서 이미지를 수집하여 보행자와 현재 차량의 주행도로를 표시한다. 이후 보행자와 차량 간의 거리를 실시간으로 측정하여 그 측정된 거리가 5m 이하일 때 바운딩 박스를 표시하며 5m 이상이라면 다시 처음으로 돌아간다. 마지막으로 현재 주행도로가 마지막 차선이고 보행자가 그 차선으로 진입할 경우 운전자에게 알람으로 위험 상황을 알린다. 추가로 현재 주행도로가 마지막 차선이 아니라더라도 보행자가 2m 이내에서 감지될 경우 운전자에게 위험 상황을 알리도록 설계하였다.

III. 실험

3.1 실험 장비 및 실험환경

3.1.1 실험 장비

제안된 논문에서 사용한 카메라는 인텔 Realsense D435 i이다. 제품의 작동 범위는 2m~10m이고, 깊이 해상도 및 FPS는 1280x720, 30FPS이다. 야외에서 실험하기 위해 NVIDIA의 Xavier를 사용했다. Xavier는 더욱 낮은 전력으로 빠르게 실행할 수 있는 높은 성능과 전력 효율을 제공한다. 또한, 카메라에 전력을 제공한다.

표 1. 사용 장비
Table 1. Equipment used

company	NVIDIA	Intel	zeuslap	Tiger Power
model name	Xavier	Realsense Depth Camera D435i	zeuslap	Tiger Power 5400mah

실험은 Xavier와 Realsense D435 i를 연결하여 실행하였다. 또한, 거리측정은 인텔에서 지원하는 pyrealsense2 를 사용했다. 또한, 전력 공급을 위해 휴대용 배터리인 Tiger power 5400mah를 사용했다. 자세한 실험 장비는 표 1에 나타내었다.

3.1.2 실험환경

그림 8과 같이 시스템을 구축하고 차량 내 설치하였다. 깊이 카메라는 차량의 중앙에 위치시켰다. 도로 환경은 2차선 도로를 가정하였다. 저속에서의 동작을 확인하기 위해 속력은 20~30km를 유지하였다. 실험은 총 3가지로 차선 인식 정확도, 보행자 인식 정확도와 사고 상황 인식 시간, 보행자 탐지 거리 정확도를 측정하였다.



그림 8. 실험환경 구축
Fig. 8. Build an experimental environment

3.1.3 실험방법

(1) 차선 인식 정확도

차량 주행 속력을 20~30km를 유지하며 중간에 차선을 1회 변경하였다. 실시간 차선 인식 결과를 저장하며 저장한 비디오를 이용해 결과를 확인하였다. 총 10회 진행을 하며 현재 주행 중인 차선을 잘 인식하는지 확인하며 주행마다 1차선과 2차선에서 각각 1회씩 캡처하여 결과를 확인하였다.

(2) 보행자 인식 정확도 & 사고 상황 인식 시간 항목 특성상 두 실험은 동시에 진행하였다. 차도에

징차한 뒤 실험을 총 10회 진행하며 영상을 저장하였다. 보행자 인식 정확도 확인 과정은 다음과 같다. 우선 보행자가 도로로 진입했을 때 경고(warning) 메시지가 출력된 경우 보행자가 인식되었는지를 확인하였다.

사고 상황 인식 시간을 확인하기 위해서 저장된 영상을 프레임 단위로 나누었다. 그 후, 보행자 도로 진입 시점부터 경고 메시지가 뜨기까지 프레임 차를 확인하였다. 실험에 사용한 카메라는 30FPS로 1초에 30프레임을 촬영할 수 있다. 이를 이용해 경고 메시지가 뜨기까지 걸린 시간을 측정하였다. 총 10회를 진행하여 보행자 인식 정확도를 확인하고 사고 상황 인식의 평균 시간을 측정하였다.

(3) 보행자 탐지 거리 정확도

자동차 안의 카메라와 보행자와의 거리를 측정하면 정확한 직선거리를 얻을 수 없다. 그러므로 테스트는 차 밖에서 진행하였다. 삼각대를 이용하여 카메라를 고정하고 뒤 줄자를 이용하여 정확히 5m 지점을 표시하였다. 총 10회 보행자가 그 지점으로 이동하며 완전히 정지한 후 거리측정 여부와 그 범위가 4.8~5.2m에 들어오는지 확인하였다. 총 10회 오차를 측정하고 평균 오차를 계산하였다.

3.2 실험 결과

3.2.1 차선 인식 정확도

그림 9는 총 2회 주행까지의 결과이다. 각각 1차선, 2차선에서의 결과를 보여준다. 또한, 차선 인식 알고리즘은 중간에 끊기지 않고 10회 주행 모두 차선 인식을 정확하게 하였다.

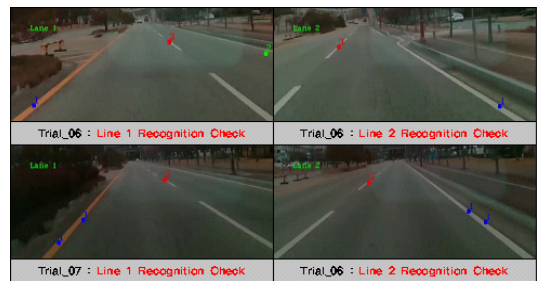


그림 9. 차선 인식 정확도 확인
Fig. 9. Check lane recognition accuracy

3.2.2 보행자 인식 정확도 및 사고 상황 인식 시간

보행자 인식 결과는 그림 10과 같다. 보행자 도로 진입 시 보행자 인식과 경고 메시지가 나타남을 확인하였다. 10회 모두 보행자가 인식됨을 확인하였다.



그림 10. 보행자 인식 및 사고 상황 인식 시간 확인
 Fig. 10. Pedestrian recognition accuracy and accident situation recognition time check

사고 상황 인식 시간의 실험 결과는 그림 10과 같다. 그림과 같이 보행자가 도로에 진입하는 시점과 경고 메시지가 출력된 프레임 차를 확인하였다. 그림에서는 도로 진입부터 인식까지 5프레임 차가 있었고, 시간상 0.1667s가 걸린 것을 확인할 수 있다. 실험은 총 10회 실행되었고, 평균은 0.1초로 확인되었다. 실험 결과는 표 2에 나타난다.

3.2.3 보행자 탐지 거리 정확도

보행자 탐지 거리측정 환경은 그림 11과 같다. 실제 거리와 측정된 거리를 비교하여 오차 범위를 확인했고, 그 평균을 구하였다. 실험 결과는 표 2에 나타난다. Trial은 시험 횟수, Meas는 측정 거리를 나타낸다.



그림 11. 보행자 탐지 거리측정 환경
 Fig. 11. Pedestrian detection distance measurement environment

표 2. 사고 상황 인식 시간 및 거리측정 시험 결과
 Table 2. Accident situation recognition time and distance measurement result

	Trial Para	1	2	3	4	5	6	7	8	9	10
		Time	5	3	2	1	2	5	5	3	3
	Time(s)	0.1667	0.1000	0.0667	0.0333	0.0667	0.1667	0.1667	0.1000	0.1000	0.1333
	Avg	0.1100s									
Dist	Meas(m)	4.98	5.15	5.03	5.11	4.94	5.11	5.03	5.07	5.11	5.15
	Error	-0.02	+0.15	+0.03	+0.11	-0.06	+0.11	+0.03	+0.07	+0.11	+0.15
	Avg	0.084m									

실험은 총 10회 진행되었다. Error는 오차 거리를 나타내고 Avg는 10회 동안 Error의 평균이다. 평균은 0.084m로 확인하였다.

IV. 결론

본 논문에서는 딥러닝을 이용한 차량과 보행자 간 사고방지 시스템을 제안하였다. 차선, 보행자 인식 및 주행도로 인식 알고리즘, 거리측정 시스템, 거리와 보행자 차선진입 여부에 따른 알람 알고리즘을 제안 및 구현하였다. 실제 주행을 통해 차선 인식 정확도, 보행자 인식 정확도, 사고 상황 인식 시간, 보행자 탐지 거리 정확도를 확인하였다. 구현한 사고방지 시스템은 저속에서도 이용 가능하므로 어린이 보호구역 등 운전자가 확인하기 어려운 시야각에 대한 위험을 줄여 주며 다방면에 적용할 수 있다.

본 논문은 2차선 도로를 기준으로 실험을 진행하였으나, 이를 기반으로 다차선 도로 환경으로 확장할 수 있다. 또한, 현재 거리측정 시스템 모델은 차체의 움직임, 역광 등 안정성 있는 거리측정에 제약점을 가진다. 따라서 이를 해결하기 위해 칼만 필터 등을 활용한 시스템 안정성 보완 확장 연구가 필요하다.

References

[1] W. Hu, et al., "Traffic accident prediction using 3-D model-based vehicle tracking," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 677-694, 2004.

[2] Z. Zhou, "Attention based stack ResNet for citywide traffic accident prediction," *2019 20th IEEE MDM*, pp. 369-370, Hong Kong, Jun. 2019.

[3] H. Zhao, et al., "Research on traffic accident

prediction model based on convolutional neural networks in VANET,” *2019 2nd ICAIBD*, pp. 79-84, Chengdu, China, May 2019.

- [4] D. C. Hernández, et al., “Lane marking detection using image features and line fitting model,” *2017 10th Int. Conf. HSI*, pp. 234-238, Ulsan, South Korea, Jul. 2017.
- [5] N. Zarbakht and J. J. Zou, “Lane detection under adverse conditions based on dual color space,” *2018 DICTA*, pp. 1-5, Canberra, Australia, Dec. 2018.
- [6] J. Feng, X. Wu, and Y. Zhang, “Lane detection base on deep learning,” *2018 11th ISCID*, pp. 315-318, Hangzhou, China, Dec. 2018.
- [7] P.-R. Chen, et al., “Efficient road lane marking detection with deep learning,” *2018 IEEE 23rd Int. Conf. DSP*, pp. 1-5, Shanghai, China, Nov. 2018.
- [8] R. Li and J. Yang, “Improved YOLOv2 object detection model,” *2018 6th ICMCS*, pp. 1-6, Rabat, Morocco, May 2018.
- [9] IntelRealSense, *revert save_to_ply options to properties(2020)*, Retrieved Nov. 25, 2019, from <https://github.com/IntelRealSense/librealsense/tree/master/wrappers/python>
- [10] S. Han, J. Park, and S. Y. Shin, “An enhanced distance measurement technique using depth camera for obstruction environment,” in *Proc. KICS Winter Conf.*, pp. 130-131, Gangwon-do, South Korea, Feb. 2020.

한 승 현 (Seung Heon Han)



2019년 2월 : 금오공과대학교 전자 공학부 졸업
 2019년 2월~현재 : 금오공과대학교 IT융복합공학과 석사과정
 <관심분야> 딥러닝, 이미지 프로세싱

[ORCID:0000-0001-9083-6773]

채 석 (Seog Chae)



1978년 2월 : 서울대학교 전자공학과 졸업
 1980년 2월 : 한국과학기술원 전기전자공학과 석사
 1989년 2월 : 한국과학기술원 전기전자공학과 박사
 1983년~현재 : 국립금오공과대학교 전자공학부 교수

<관심분야> 지식 기반 지능 시스템, 네트워크 기반 제어

박 재 한 (Jae Han Park)



2020년 2월 : 금오공과대학교 전자공학부 졸업
 2020년 2월~현재 : 금오공과대학교 IT융복합공학과 석사과정

<관심분야> 딥러닝, 이미지 프로세싱

[ORCID:0000-0002-3856-5846]

사이드 알리 하산 (Syed Ali Hassan)



2016년 : Mohamand Ali Jinnah University BS Software Engineering 졸업

2018년~현재 : 금오공과대학교 IT융복합공학과 석사과정

<관심분야> AR/VR/MR and Deep Learning.

[ORCID:0000-0002-5872-1615]

라힘 타릭 (Tariq Rahim)



2012년 9월 : COMSATS Institute of Information Technology Electronics Engineering 졸업
2017년 7월 : Beijing Institute of Technology Information and Communication Engineering 석사

2017년 7월~현재 : 금오공과대학교 IT융복합공학과 박사과정

<관심분야> image and video processing and quality of experience (QoE) for high resolution and high frame rate videos

[ORCID:0000-0001-7817-9715]

신 수 용 (Soo Young Shin)



1999년 2월 : 서울대학교 전기공학부 졸업
2001년 2월 : 서울대학교 전기공학부 석사
2006년 2월 : 서울대학교 전기공학부 박사
2010년~현재 : 국립금오공과대

학교 전자공학부 교수

<관심분야> 5G/B5G 무선 접속 기술, 드론 응용, 혼합 현실, 블록체인, 머신러닝 및 딥러닝

[ORCID:0000-0002-2526-2395]