



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

공학전문석사 학위 연구보고서

자율주행 환경에서 운전자 측면 얼굴의
졸음 실시간 감지 시스템

**Real-time Eye Blink Detection System for Driver's
Side Face in Autonomous Driving Environment**

2023년 2월

서울대학교 공학전문대학원
응용공학과 응용공학전공
구본학

자율주행 환경에서 운전자 측면 얼굴의 졸음 실시간 감지 시스템

Real-time Eye Blink Detection System for Driver's Side Face in Autonomous Driving Environment

김성우

이 프로젝트 연구보고서를 공학전문석사 학위
연구보고서로 제출함

2023년 2월

서울대학교 공학전문대학원
응용공학과 응용공학전공
구본학

구본학의 공학전문석사 학위 연구보고서를

인준함

2023년 2월

위원장 전 홍 범 (인)

위 원 성 원 용 (인)

위 원 김 성 우 (인)

초 록

최근 자율주행 2단계에 자동차들이 많이 나오기 시작하면서 운전자들은 운전 집중하기보다는 자동차의 엔터테인먼트 시스템으로 시선을 옮겨가고 있고, 자율 주행 시스템의 도움으로 운전이 단조로워지면서 졸음에 빠지는 경우가 많아지고 있다. 하지만 아직까지 완전 자율 주행의 단계까지는 여러가지 단계들이 남아있다. 이러한 중간 시점에서 우리는 운전자들의 집중할수 있게 하거나 졸음에서 깨어날수 있도록 도와주는 시스템이 필요하다.

지금까지의 연구들은 자율 주행 환경이 아닌 일반 주행 환경에서 운전자가 시선을 앞만 보고 있을때에 졸음을 감지하는 방법에 대해서 집중해서 연구해왔다. 이에 따라서 본 논문에서는 비접촉식 방식중의 하나로 카메라를 통해서 운전자가 엔터테인먼트 시스템으로 시선을 옆으로 두거나 졸음을 했을때 운전자에게 주의를 줄수 있는 자율환경에서 운전자 졸음 방지 시스템을 연구하여 개발하였다.

졸음 운전을 판단하기 위해서 먼저 운전자의 얼굴을 인식하고, 얼굴의 주요지점을 인식한 이후에 필요한 눈과 입의 주요지점을 찾아내고, 찾아낸 주요지점을 통해서 머리의 자세까지 판단하였다. 그리고 현재 많은 시스템에서 잘 동작하지 않고 있는 고개를 옆으로 돌렸을때에 본 논문에서 제시한 알고리즘을 통해서 눈의 크기를 보정하도록 시스템을 설계하였다.

본 논문에서 python 개발 환경에서 opencv를 주로 사용하였다. 감지 모델로는 얼굴을 인식하고 얼굴 주요지점을 인식하는 방법으로는 높은 성능을 내는 MobileNet SSD 를 통해서 얼굴을 감지하였고, 얼굴 주요 지

점 감지 모델로는 Regression Tree 방식을 이용하여 빠른시간에 얼굴 주요 지점을 감지하였고, 머리의 자세는 앞에서 찾은 얼굴 지점의 2D 좌표를 이용하여 Perspective-n-Point 방법을 사용하여 얼굴의 회전 방향을 추측하였고, 그를 활용하여 옆 얼굴에서 눈의 크기를 보정하는 방법을 개발하여 전체적인 시스템 개발을 하였다.

본 연구를 통해 자율 주행 2단계 환경에서 운전자가 옆으로 봤을때에 졸음을 이전 시스템들보다 더 잘 감지할수 있음을 확인하였다. 본 연구 결과를 실무 환경에 적용함으로써 운전자의 졸음을 더 정확히 판단할 것을 기대하며 이는 소프트웨어 경쟁력을 확보하는데 기여할 것이다.

주요어 : 자율주행, 실시간, 졸음감지

학번 : 2021-26571

목 차

I. 서론	1
1.1 연구의 배경과 목적	1
1.2 연구 보고서의 구성	6
II. 관련 연구	7
2.1 EuroNCAP 안전관련 평가에 대한 연구	7
2.2 자율주행단계	11
2.3 졸음 운전자 감지를 위한 방법들	14
2.4 적외선 센서를 이용한 눈 검출방법	18
2.5 대표적인 얼굴 감지 모델	18
2.6 기존연구와의 차이점	19
III. 졸음 감지 기법 시스템	22
3.1 얼굴 감지	24
3.2 얼굴 주요지점 감지	25
3.3 머리 자세 감지	26
3.4 눈 크기 교정	27
3.5 눈깜빡임 감지	28
3.6 하품 감지	31
3.7 졸음 감지를 위한 변수들	32
IV. 실험 결과	33
4.1 실험환경	33

4.2	얼굴 감지 모델 성능 비교	35
4.3	뜬눈의 EAR과 감은눈의 EAR 차이 실험	36
4.4	EAR 감지기 성능 비교 실험	37
4.5	눈 깜빡임 성능 실험	38
V.	결론	40
5.1	연구 결과	40
5.2	후속 연구 과제	41
	참고 문헌	42
	Abstract	45
	찾아보기	47

그림 목 차

그림 1.	수동적인 평가	1
그림 2.	능동적인 평가	2
그림 3.	GUI 없는 단순 라디오 시스템	3
그림 4.	GUI가 있는 스마트 시스템	4
그림 5.	시선의 변화	5
그림 6.	Seat Belt Reminder	7
그림 7.	Adaptive Cruise Control	8
그림 8.	Lane Keep Assist	9
그림 9.	Autonomous Emergency Breaking	9
그림 10.	Driver Monitoring System	10
그림 11.	SAE's Automation driving level	12
그림 12.	Biometric information	16
그림 13.	Vehicle Behavior	17
그림 14.	Driver's graphic information	17
그림 15.	졸음 감지 시스템 구성도	23
그림 16.	HoG&SVM 모델과 비교	25
그림 17.	Facial landmarks와 head pose 예측	27
그림 18.	머리 자세에서 구한 Yaw, Roll, Pitch	28
그림 19.	EAR을 위한 Point	29
그림 20.	Frame 에 따른 EAR 변화	30
그림 21.	MAR을 위한 Point	31
그림 22.	Bench Test 환경	34

그림 23.	정면에서 EAR 성능 비교	37
그림 24.	측면에서 EAR 성능 비교	38
그림 25.	머리자세에 따른 EAR보정 결과	38

표 목 차

표 1.	졸음운전자 감지를 위한 방법들	15
표 2.	기존연구와의 차이점	21
표 3.	일반적인 얼굴 모델의 얼굴 포인트의 3D 좌표	26
표 4.	카메라 정보	26
표 5.	HOG&SVM 해상도에 따른 수행 시간	35
표 6.	MobileNet SSD v2 해상도에 따른 수행 시간	36
표 7.	뜬눈의 EAR값고 감은 눈의 EAR 실험	36
표 8.	얼굴 정면 눈 깜빡임 성능	39
표 9.	얼굴 측면 눈 깜빡임 성능	39

제 1 장

서론

1.1 연구의 배경과 목적

수십년 동안 자동차의 안전에 대한 평가는 [그림.1] 과 같이 충돌이 발생했을 때 자동차가 얼마나 승객을 안전하게 보호할 수 있는지에 대해서 수동적인 관점 중심으로 평가해오고 있었다.

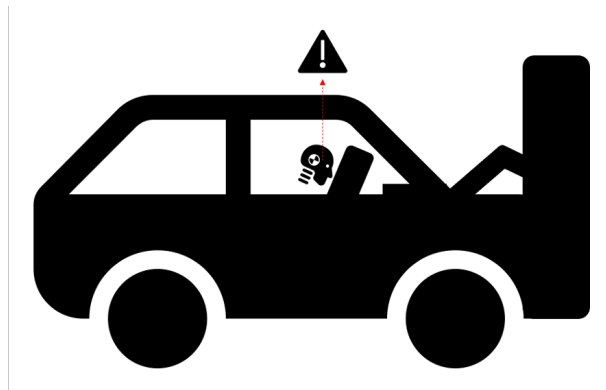


그림 1: 수동적인 평가

하지만, 최근 10년동안은 자동차가 사고가 발생하기전에 [그림.2] 와 같이 능동적으로 운전자에게 위험을 알리고 사고가 발생하지 않도록 도움을 주는지에 대해서 평가를 해오고 있다.

우리나라의 교통사고 원인 별 사망자 현황은 한국도로공사의 통계에 따르면 과속, 무단횡단, 차량결함 보다는 졸음 및 주시태만으로 인한 사고원인이 67%로 가장 크다.

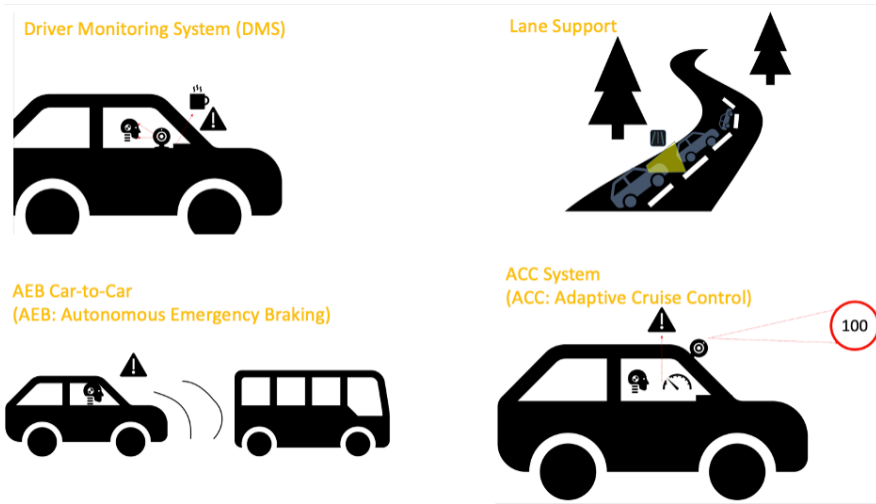


그림 2: 능동적인 평가

이러한 배경으로 Euro New Car Assessment Program (EuroNCAP)에서는 2020년 부터는 Driver Monitoring System (DMS)를 평가하기 시작하면서 수동적인 관점뿐만 아니라 능동적인 관점에서 안전의 지표를 추가하여 평가를 하고 있다. 그리고 최근 자율주행 크게 발전하면서 자율주행 기능을 탑재한 자동차들이 시장에 많이 나오고 있다. 하지만 아직까지는 자율주행 자동차들은 운전자의 개입이 필요 없는 완전 자율 주행이 아니라 운전자가 의식을 가지고 운전 집중을 해야 하는 자율 주행 2단계 정도에 머무르고 있다.

그리고 자동차들의 엔터테인먼트 시스템은 10년전에는 [그림.3]과 같이 GUI가 없는 대부분의 라디오나 CD Player 또는 Bluetooth 기능으로 소리만 들려 주는 시스템이 대부분이었고 터치스크린이 아닌 대부분 하드키로 입력을 할수 있는 시스템이었다.

하지만 2010년 부터 스마트폰의 발전으로 소비자들은 GUI 환경에 익숙해지면서 자동차에서도 [그림.4]과 같이 Human Machine Interface



그림 3: GUI 없는 단순 라디오 시스템

(HMI)가 있고 터치스크린을 지원하는 시스템이 시장에서 인기를 얻게 되었고 이를 지원하기 위해서 6인치 정도의 작은 디스플레이가 있는 제품이 시장에 출시되기 시작하였다. 그리고 최근에는 이러한 엔터테인먼트 시장이 빠르게 발전하면서 테슬라사에서는 15인치 디스플레이를 출시하고 있다.

이러한 엔터테인먼트의 발전으로 인해 운전자들은 정면을 바라보고



그림 4: GUI가 있는 스마트 시스템

운전을 하기 보다는 [그림.5]과 같이 발전한 엔터테인먼트 시스템에 시선을 오랫동안 두게 되었다.

이러한 문제에 추가적으로 최근에 발전한 자율 주행 기능을 사용함으로써 운전자들은 운전 집중하기 보다는 잠을 자거나 엔터테인먼트 시스템에 집중하는 문제들이 더욱 더 두드러지고 있다.

최근에는 테슬라 자율주행 자동차에서 졸면서 운전을 하는 뉴스가

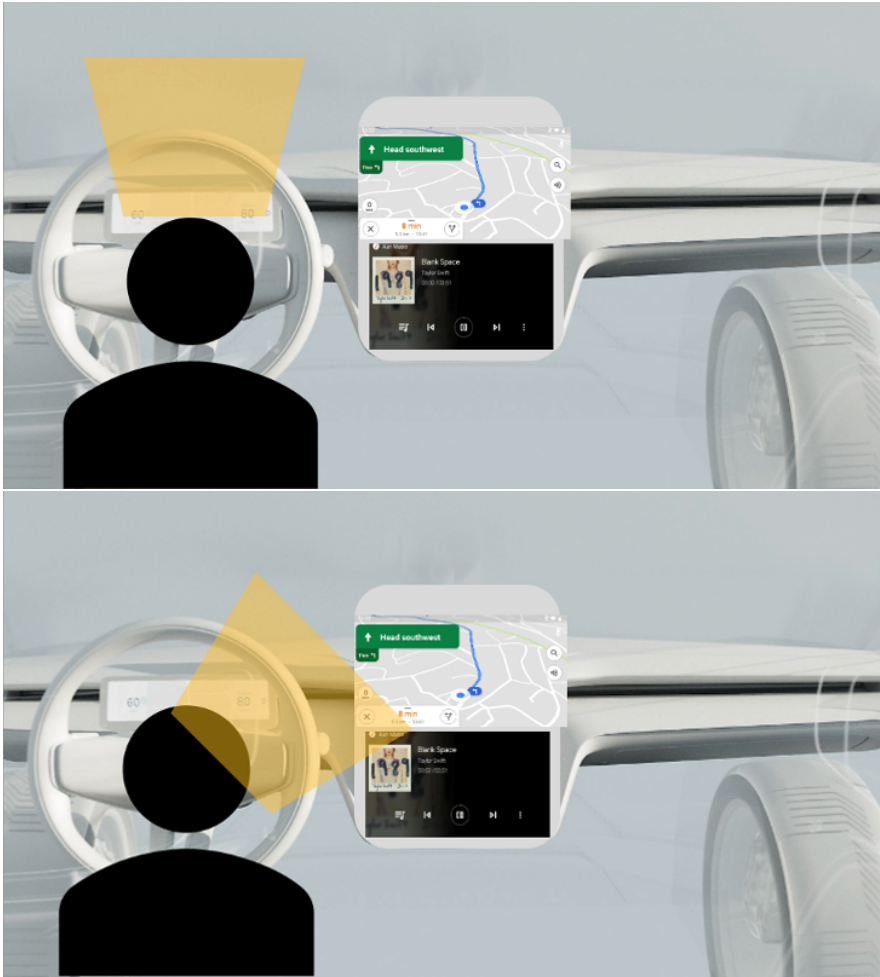


그림 5: 시선의 변화

나와서 사회적 큰 파장을 일으키기도 했다. 하지만 아직까지 완전자율 주행의 단계까지 발전하기에는 여러가지 넘어야하는 단계들이 많이 남아 있다.

이러한 4가지 연구 배경으로 자율 주행환경에서 졸음 하거나 운전자가 정면을 주시하고 운전하지 않고 발전한 엔터테인먼트에 시선을 돌리는 주의 태만을 감지하는 시스템이 필요하다. 하지만 현재까지 나와 있는 시

스택과 연구들은 자율 주행 환경에서 발전한 인포테인먼트 시스템으로 운전자의 시선이 많이 돌아가고 있지만, 여전히 얼굴의 측면에서의 졸음 감지는 어려움을 가지고 있다. 이러한 측면에서 자율 주행 환경에서 실시간 졸음 감지의 어려운 문제점을 본 연구에서는 개선하는 시스템에 대해서 연구를 진행하고자 한다.

1.2 연구 보고서의 구성

본 연구 보고서의 구성은 다음과 같다. 제 1장에서는 이미 설명한 것처럼 연구의 배경과 목적에 대해서 기술하고 있다. 제 2 장에서는 본 연구에 앞서 필요한 관련 연구들에 대해서 제시하고 있다. 첫 번째로 EuroNCAP 안전관련 평가에 대한 연구를 진행하고, 두 번째로 자율 주행 단계에 대해서 간략하게 설명한다. 세 번째로는 운전자를 모니터링하는 카메라들의 추세에 대해서 조사하였다. 네 번째로는 졸음 운전자를 감지하기 위한 방법들에 대해서 조사하였다. 다섯 번째로는 졸음 감지를 위한 변수들에 대해서 조사하고 기술하였다. 여섯 번째로는 적외선 센서를 이용한 눈 검출방법에 대해서 조사하였고 마지막으로 기존 연구들과 본 연구의 차이점에 대해서 제시한다. 제 3 장에서는 본 연구가 제시하고자 하는 졸음 감지 기법 시스템에 대해서 단계별로 하나씩 기술하면서 설명한다. 제 4 장에서는 졸음 감지 기법 시스템의 실험의 환경, 방법과 실험 결과에 대해서 기술한다. 마지막으로, 제 5장에서는 본 연구 보고서에서 기술한 연구 내용을 정리하고 후속 연구 과제와 함께 기술하며 끝 맺는다.

제 2 장

관련 연구

2.1 EuroNCAP 안전관련 평가에 대한 연구

EuroNCAP에서는 2000년도 이전에는 충돌 사고가 발생했을 때에 에어백을 통해서 부상률을 낮추는데 중점적으로 평가를 해왔다. 그리고 2000년 초반에는 Occupant Status Monitoring (OSM) System의 [그림.6] SeatBelt Reminder(SBR)를 평가하기 시작했으며, 2000년 후반에는 운전

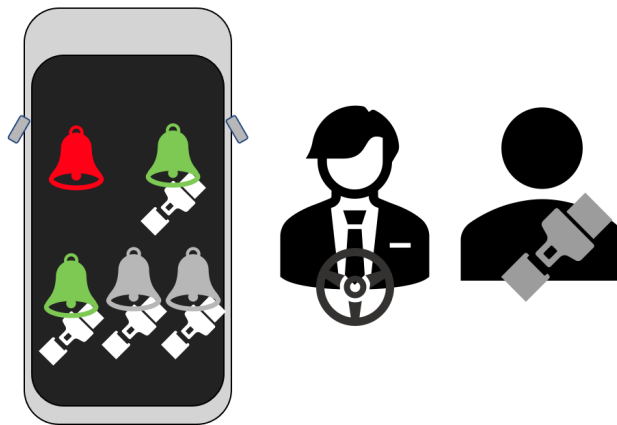


그림 6: Seat Belt Reminder

자가 차를 제어할 수 있도록 돕는데 효과적인 Electronic Stability Control (ESC)가 중요한 평가의 요소가 되었고, 2014년에는 ESC 장착이 의무화

되었다. 또한 2000년 후반에는 Speed Assistance가 중요시되기 시작하여 Speed Limiter (SL), [그림.7] Adaptive Cruise Control (ACC) 와 같은 시스템을 결합하여 운전자가 주의를 흐뜨리지 않는지를 확인하였다.

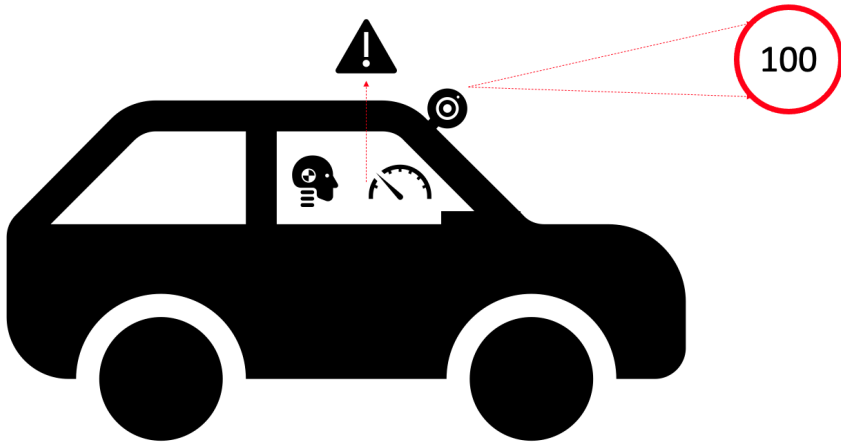


그림 7: Adaptive Cruise Control

그리고 2014년부터는 Lane Support 가 중요한 평가요소가 되어, 자동차가 차선을 벗어날 경우 차량의 경로를 수정하는데 도움을 주는 [그림.8] Lane Keep Assist (LKA) 시스템이 안전관련 평가에 추가되었다. 같은 시기인 2014년에 센서를 사용하여 차량 전방에 잠재적인 위험의 존재를 감지하고 운전자가 제시간에 위험을 감지하지 않을 경우 충돌을 피하거나 심각성을 완화하기 위한 [그림.9] Autonomous Emergency Breaking (AEB) 기술이 중요한 평가 요소가 되면서 센서 기반의 안전 보조 시스템이 중요시되기 시작하였다. 최근 2020년부터는 운전자의 주의 산만 및 피로에 대해서 모니터링 하는 [그림.10] Driver Monitoring System (DMS) 을 평가하기 시작하면서, 본 연구에서 소개하는 운전자 집중도 위험 감지가 필요하게 되었다.

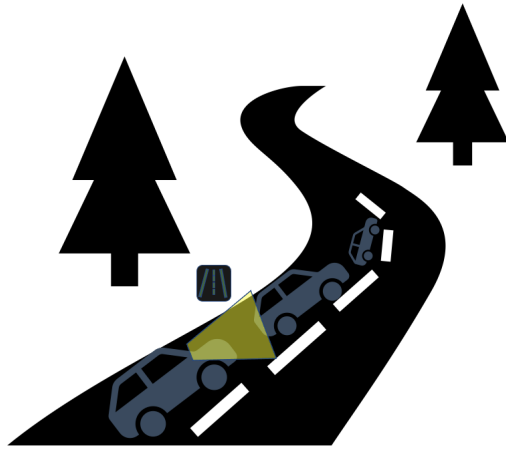


그림 8: Lane Keep Assist

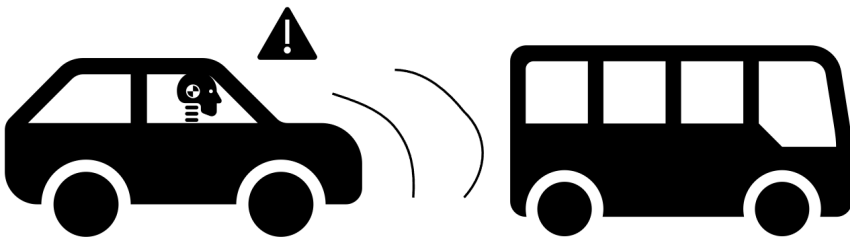


그림 9: Autonomous Emergency Braking

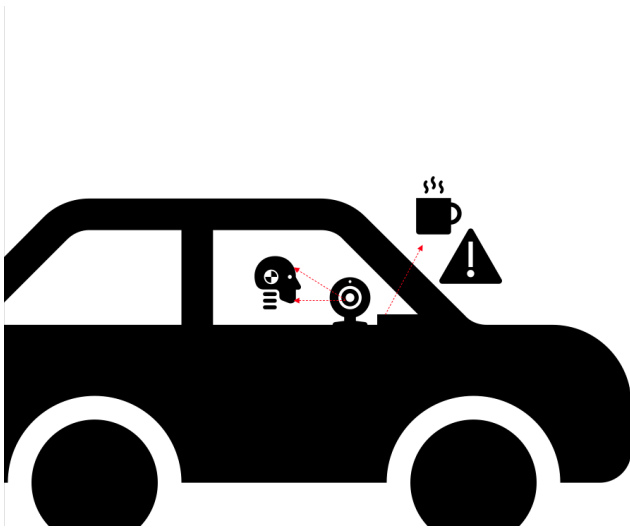


그림 10: Driver Monitoring System

2.2 자율주행단계

Society of Automotive Engineering (SAE) 에서 2014년 이후에 자동차용 운전 자동화 시스템과 관련된 용어에 대한 분류 및 정의를 그림11과 같이 하고 있다.



	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
레벨 구분						
	운전자 보조 기능			자율주행 기능		
명칭	비자율주행 (No Automation)	운전자 지원 (Driver Assistance)	부분 자동화 (Partial Automation)	조건부 자동화 (Conditional Automation)	고도 자동화 (High Automation)	완전 자동화 (Full Automation)
자동화 항목	없음(경고 등)	조향 or 속도	조향 & 속도	조향 & 속도	조향 & 속도	조향 & 속도
운전 주시	항시 필수	항시 필수	항시 필수 (조향핸들 상시 잡고 있어야함)	시스템 요청시 조향핸들 잡을필요X, 제어권 전환시만 잡을필요 O	작동구간 내 불필요 (제어권 전환X)	전 구간 불필요
자동화 구간	-	특정구간	특정구간	특정구간	특정구간	전 구간
예시	사각지대 경고	차선유지 또는 크루즈 기능	차선유지 및 크루즈 기능	혼잡구간 주행지원 시스템	지역 무인택시	운전자 없는 완전자율주행

그림 11: SAE's Automation driving level

- Level 0 : 자율주행 없음 - 자동차가 운전자의 의해서 제어 되어야 하는 레벨.
- Level 1 : 자율주행보조 - 자동차가 하나의 자율운전 시스템을 갖추고 있는 레벨로 클루저 컨트롤과 같은 속도조절 같은 기능을 갖추고 있다.
- Level 2 : 부분적인 자율주행 - 운전자가 자동차에게 속도조절과 운전대 제어를 맡기는것을 허락 하는 레벨
- Level 3: 조건적인 자율주행 - 자동차가 환경적인 요소를 감지하는 기능을 가지고 있어서 자동차 스스로 자신의 결정을 내릴수 있는 레벨이지만 시스템이 그러한 임무를 수행하지 못한다면 운전자는 반드시 자동차를 제어하고 경계해야하는 레벨
- Level 4: 고도의 자율주행 - 대부분의 경우 인간의 개입을 필요하지 않다. 하지만 인간에 의해서 제정비 될수 있는 레벨
- Level 5: 완전 자율 주행 - 인간의 주목이 필요없는 상태로 인간이 운전해서 갈 수 있는 어디든 갈수 있는 레벨

졸음 감지를 위한 시스템은 현재 SAE Level 2에 머물러 있는 현상황에서는 필요하고 SAE Level 3에서도 운전자가 개입이 필요하지만 Level 올라가면 갈수록 운전자는 운전에 집중을 하지 않게 되고 이러한 단조로운 운전은 결국 운전자를 졸음으로 빠뜨릴수 있는 확률이 높아진다. 이런 상황에서 더욱더 졸음 감지 시스템이 잘 발달되어 운전자가 졸지 않도록 시스템을 구현하는것이 필요하다.

2.3 졸음 운전자 감지를 위한 방법들

졸음 운전을 하는것은 자율주행과 같은 운전자의 각성 상태가 떨어지는 조건과 단조로운 운전하는 조건과 다른 환경적인 요소들에 의해서 발생할수 있다. 졸음운전을 감지하는 문제를 풀기 위한 중요한 요소는 얼마나 정확한지와 이른 시기에 감지 할 수 있는가 중요하다[1].

표 1: 출음운전자 감지를 위한 방법들

Methods	Measurement Information	Measurement Method
Contact method	Biometric information	Heart rate monitor, pulse wavemeter, electroencephalograph, electromyograph, nystagmus, etc.
Non-contract method	Vehicle behavior	steering pattern, distance between lane and vehicle speed, distance between vehicles, etc.
	Driver's graphic information	Open rate of eyes, blink, pupil, voice, and express

졸음 운전자 감지를 위한 방법들은 측정 정보의 분류로 3가지로 구분된다[2]. 첫번째는 Biometric information 방법으로 심장박동, 뇌파, 맥박을 이용하는 것으로 Electrocardiogram (ECG), electroencephalogram(EEG), electrooculogram(EOG), electromyography(EMG)를 이용하여 측정하는 방식인데 실제 차량에 적용하는 것은 현실적으로 어렵다[3].

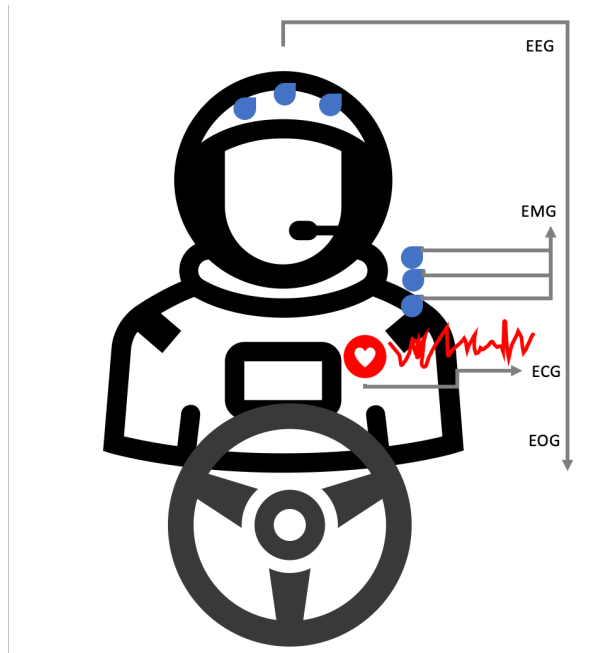


그림 12: Biometric information

두번째는 [그림.13]Vehicle Behavior를 통한 방법으로 Steering Wheel Pattern 분석 및 차와 차선의 거리 측정, 속도측정, 앞차와의 간격을 통해서 측정하는 방법이 있는데 해당 방법은 도로 환경에 의존성이 크고, 운전자가 steering wheel을 잡고 있을 때에만 유용하다.

마지막 방법은 [그림.14] Driver's graphic information을 활용하는 방법으로 눈의 깜빡임, 눈이 떠있는 비율을 통한 방법으로 높은 정확도를

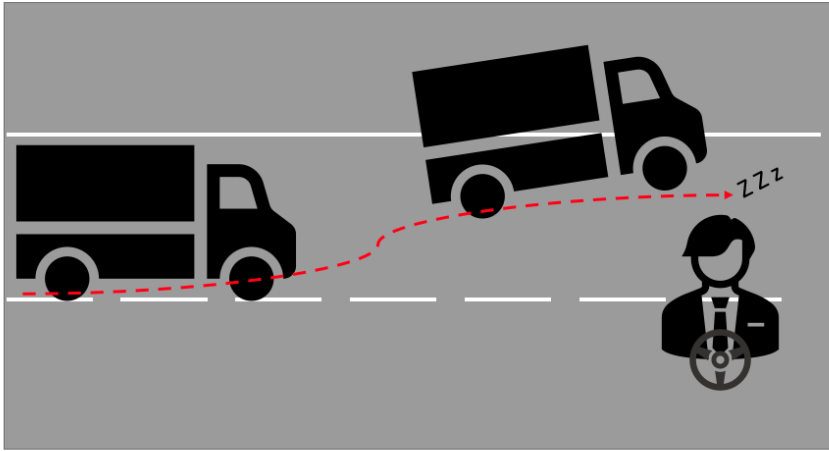


그림 13: Vehicle Behavior

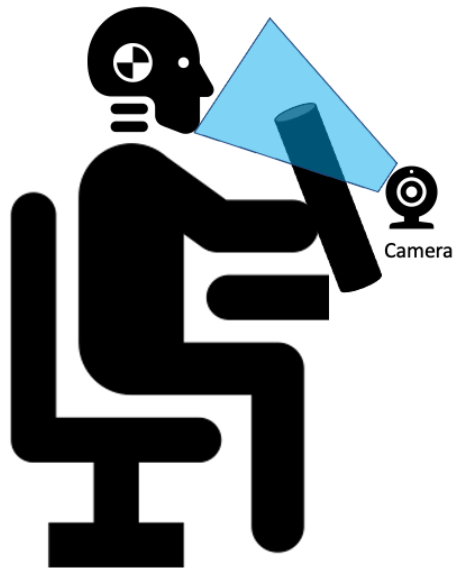


그림 14: Driver's graphic information

가지지만, 빛의 환경에 따라서 performance가 떨어지고 운전자가 고개를 돌렸을 때 정확도가 떨어진다.

2.4 적외선 센서를 이용한 눈 검출방법

현재 개발되어 있는 시스템에서는 적외선 센서를 이용하여 적외선 조명을 교차로 비추어 각막 반사 현상을 통해서 White pupil 과 Black pupil의 명암 차를 이용하여 두 눈의 위치를 검출하는 방법을 사용하고 있는 시스템이 있다 [4][5]. 이 방법은 안경을 쓴 사람에게도 견고하게 잘 동작한다. 하지만, 본 논문에서 진행하려고 하는 것은 RGB 카메라에서 얼굴과 눈 이외의 하품을 감지하기 위함 입과 머리의 포즈를 감지할수는 없기 이유로 사용하지 않았다.

2.5 대표적인 얼굴 감지 모델

졸음 운전자 감지를 위해서 우선적으로 해야하는 연구는 얼굴을 인식하는것이다. 얼굴을 감지하는 모델은 크게는 두가지로 분류할수 있다. 첫번째는 머신러닝방식 모델이고, 두번째로는 딥러닝방식 모델이 있다.

Computer Vision 학문에서는 오래전부터 얼굴을 감지하는 방법에 대해서 연구를 해왔고, 머신러닝 방식으로 가장 유명한 2개의 방법은 2001년의 P. Viola가 제안한 Haar Cascades Classifier[6] 모델과 N. Dalal가 2005년에 제안한 Histogram of Gradients[7] 와 Support Vector Machine (SVM) 을 사용하는 방법이 있다.

P. Viola의 방법은 다수의 객체 이미지와 객체가 아닌 이미지를 cascade 함수로 트레이닝 시켜 객체 검출을 달성하는 머신러닝 기법의 접근 방법이다.

N. Dalal 의 방법은 Edge의 Gradient가 얼마나 크지와 어느 방향인지를 구분하는 feature를 가지고 있는것이 특징이다. 이 특징을 가지고 SVM 분류기로 얼굴을 분류하는 방법이다.

이후 딥러닝 방식들이 소개되었는데, 얼굴 감지에서 유명한 방식들은 3가지의 방법이 있다. 그중 첫번째는 2012년에 A. Krizhevsky의 Convolutional Neural Networks (CNN)[8] 모델이다. 이 방법은 Convolution이라는 전처리 작업이 들어가 있는 Neural Network 모델입니다. 이미지를 처리하면서 한줄 데이터로 만들면서, 이미지의 공간적/지역적 정보가 손실되게 된다. 이러한 문제점을 극복하기 위해서 이미지를 날것 그대로 받으므로써 공간적/지역적 정보를 유지한 채 특징들의 계층을 만드는 방식이 CNN의 특징이다.

다음 방법으로는 Multi-task Cascaded CNN (MTCNN)[9]이다. 2016년에 Zhang이 Signal Processing 에서 소개하였고 핵심아이디어는 얼굴 인식과 얼굴 정렬 두가지 테스트가 연결되어 있을것이라는 아이디어로 얼굴 위치를 나타내는 박스의 위치를 세밀하게 조절해주는 Bounding box regression 테스트를 추가하여 3가지 테스트를 수행하는것이다.

마지막은 Y. Chen의 Mobilenet-SSDv2[10] 방법이다. 해당 방법은 모바일에서 돌릴수 있도록 경량화한 Single shot multibox detector이다. 본 연구에서는 Mobilenet-SSDv2 모델을 사용하였다.

2.6 기존연구와의 차이점

운전자의 졸음 방지를 위한 연구들은 지금까지 많이 진행되어 왔다. Zhang et al[11]. 은 운전자의 얼굴을 Haar을 통해서 감지하고 Boost-LBP (Local Binary Pattern) 과 SVM(Support Vector Machine)을 통해서 얼굴의

특징점을 감지하였다. 하지만 LBP를 통해서 얼굴의 특징점을 표정으로 분류하고 있고 연속적인 비디오가 아니라 한장의 사진으로 진행하였다. Picot et al.[12].은 Electro-oculogram(EOG) 장비를 이용하여 현실적으로 사용하기 어렵다는 한계를 가지고 있다. Akrou et al.[13]은 처름으로 머리 자세 추정을 도입했지만, 실제적으로 -20° 부터 20° 까지만 고려하여 해당 각도 이상되면 경고를 울리는 시스템이다. Wanghua et al.[14]은 MC-KCF(Multiple Convolutional Neural Networks-Kernelized Correlation Filters)를 통해서 처음으로 Deep Learning 방식을 채택했지만 머리자세추정을 하지 않았고 서버와통신을 해야하는 한계가 있다. Giovanni et al.[15]은 실시간 감지와 정면 얼굴의 인식이 잘되는 장점을 가지고 있지만 얼굴을 옆으로 보고 있을때의 감지는 잘되지 않는 한계를 가지고 있다. 본 논문에서 제안하는 방법은 실시간으로 감지를 하고 있고 얼굴감지를 Deep learning 방식으로 하고 있어서 정확도가 높고 자율주행환경에서 측면을 보고 있을때에도 높은 인식율을 가지는 장점을 가지고 있다.

표 2: 기존연구와의 차이점

	Zhang[11]	Picot[12]	Akrout[13]	Wanghua[14]	Giovanni[15]	Ours
얼굴 감지	Haar	Haar	Haar	MC-KCF	HOG&SVM	MobileNet-SSDv2
얼굴 특징점 감지	Boost LBP +SVM	HoG+OPL filter	Circular Hough Transform	Cascade of regression	Cascade of regression	Cascade of regression
머리 자세 추정	X	X	O	X	O	O
실시간성	X	X	X	O	O	O
얼굴 옆면 감지	X	X	X	X	X	O
특징	연속적인 비디오 사진으로 탐지 머리포즈 고려 X	장비 착용필요 얼굴 인식을 낮춤 실시간성 X	$\pm 20^\circ$ 이상 경고	얼굴추적중점 서버 통신 필요 얼굴 측면 고려 X	정면 얼굴 인식을 낮춤	측면까지 고려됨

제 3 장

졸음 감지 기법 시스템

본 논문에서는 졸음 감지 시스템을 개발하기 위해서 비접촉식 방법 중에 운전자의 그래픽 정보를 이용한 방법을 사용하였다. 해당 시스템의 개발은 Python, OpenCV와 Tensorflow를 사용하여 개발하였다. [그림.15]은 카메라를 통해서 운전자 졸음 상태를 감지하기 위한 시스템 구성도를 나타낸다.

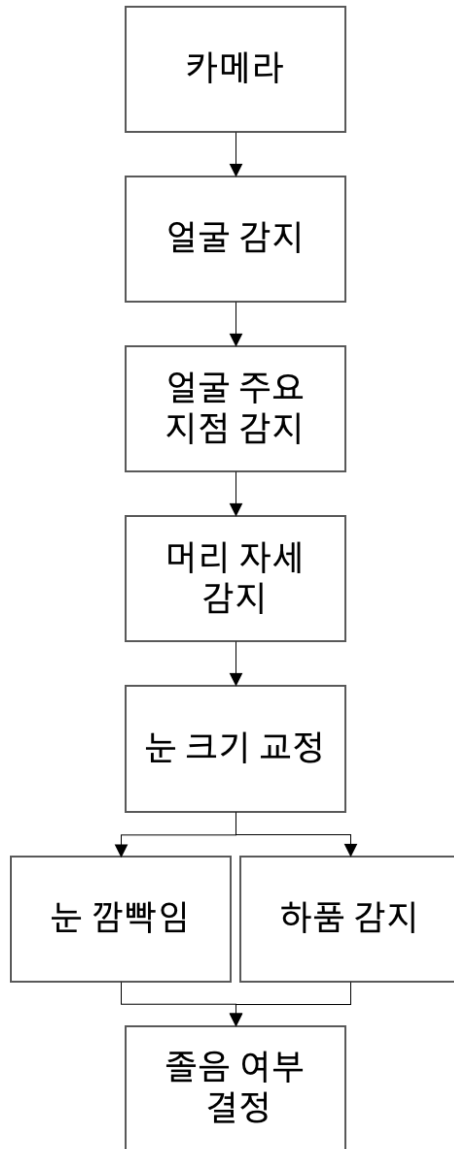


그림 15: 졸음 감지 시스템 구성도

처음 단계로 카메라를 통해서 프레임을 읽어오고, 해당 프레임에서 얼굴을 먼저 감지한다. 얼굴을 감지는 방법으로 MobileNet-SSDv2[10]을 이용하여 감지 하였다. 한개의 프레임에서 얼굴을 인식한 관심영역만 다시 얼굴의 주요지점을 찾기위해서 입력으로 넣고 얼굴의 주요지점을 감지 하였다. 얼굴 주요지점을 찾는 방법에서도 Cascade Regression[16] 모델을 사용하여 주요지점을 감지하였다. 그리고 머리 자세를 Perspective-n-Point (PnP) 방식으로 예측한다. 그 이후에 본 논문에서 다른 연구들과 다르게 얼굴의 주요지점을 감지한것에서 눈과 입의 크기 비율을 계산하고 머리 자세에서 얻어진 머리의 각도를 이용해서 눈이 깜빡인것인지 보정 하도록 알고리즘을 설계하였다. 그 이후에 눈의 깜빡임을 감지하고 하품을 감지하도록 하였다. 마지막으로 졸음 여부를 결정하는 몇가지의 변수들을 고려해서 최종 졸음 여부를 판단하도록 하였다.

3.1 얼굴 감지

본 연구에서는 선행연구를 통해서 조사했던 여러가지 방법들을 실험을 비교했고, 최종적으로 가장 좋은 성능을 가지고 있는 MobileNet SSD v2[10]를 사용하였다. MobileNet SSD v2방식은 구글에서 내놓은 경량 모바일 딥러닝 모델에서 이전에 있었던 2 단계의 감지기가 아닌 Single-Shot Multibox Detector (SSD) 로 얼굴을 감지하는것과 얼굴의 위치를 찾는것을 두 단계가 아닌 한 단계로 감지하는 모델인데, 두 단계의 모델들보다 학습 속도가 빠르다. 본 연구는 임베디드 시스템에 적용하기 위한 실시간 감지기이므로 가볍고 빠른 모델을 선택하였다. 해당 모델의 데이터셋은 WIDERFace[17] 을 사용하여 32,203개 이미지, 293703개의 얼굴 레이블이 존재한다. 성능 측면에서는 mAP 가 0.751, Average IOU는 0.598 로

성능이 우수하다. 선행연구에서 설명했던 머신러닝 기반의 얼굴 감지기인 HOG&SVM 모델을 사용했을 때에는 [그림.16]과 같이 얼굴을 잘 감지 못하는 것을 볼 수 있지만, 본 연구의 방법으로는 얼굴을 잘 인식 하는 것을 알 수 있다.

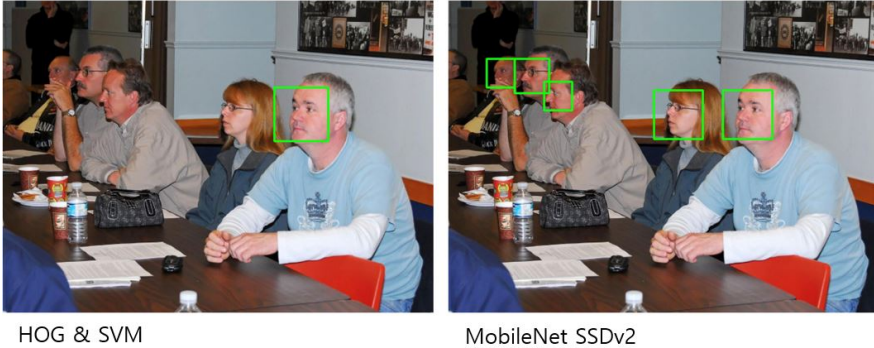


그림 16: HoG&SVM 모델과 비교

3.2 얼굴 주요지점 감지

얼굴의 주요지점을 검출하는 방법으로는 V. Kazemi의 Cascade of Regression Functions[16]을 사용하여 앞에서 감지된 얼굴영역을 입력으로 넣고 총 68개의 주요지점을 출력으로 받는 모델을 사용하였다. 얼굴의 주요지점 감지는 눈, 눈썹, 코, 입, 턱선등을 지역화 해서 나타내는 것입니다. 여기에서 눈의 위치를 정확히 감지해서 이후의 눈을 감고 있는지 뜨고 있는지에 대해서 알 수 있습니다.

3.3 머리 자세 감지

머리 자세 감지는 Computer Vision 학문에서 Perspective-n-Point (PnP) 문제로 분류하고 있습니다. 머리 자세를 구하기 위해서는 3가지를 알아야 합니다. 첫번째로는 2D 좌표입니다. 2D 좌표는 카메라로 부터 읽어들인 프레임이 2D로 구성이 되고 그 프레임에서 이전에 감지한 얼굴 주요지점들의 x,y 좌표를 통해서 2D 좌표를 구할수 있습니다. 두번째로는 3D 좌표입니다. 3D 좌표는 일반적으로 [표.3] 처럼 정의하여 사용 합니다. 마지막으로 카메라의 정보가 필요합니다. 카메라의 매개변수들인 Focal Length와 Camera 중심점과 왜곡 상수들이 필요합니다. 해당 변수들은 [표.4]와 같이 정의되어져 있습니다. 이러한 3개의 정보가 있으면 이동을 위한 Translation 값과 회전을 위한 Rotation 값을 [그림17]과 같이 구할수 있습니다.

표 3: 일반적인 얼굴 모델의 얼굴 포인트의 3D 좌표

3D 좌표	얼굴 포인트
(0.0, 0.0, 0.0)	코끝
(0.0, -330.0, -135.0)	턱
(-225.0, 170.0, -135.0)	왼쪽 눈 가장자리
(225.0, 170.0, -135.0)	오른쪽눈 가장자리
(-150.0, -150.0, 125.0)	입의 왼쪽가장자리
(150.0, -150.0, -125.0)	입의 왼쪽가장자리

표 4: 카메라 정보

Focal length	640
Camera center	(320,240)
Distortion 상수	없음

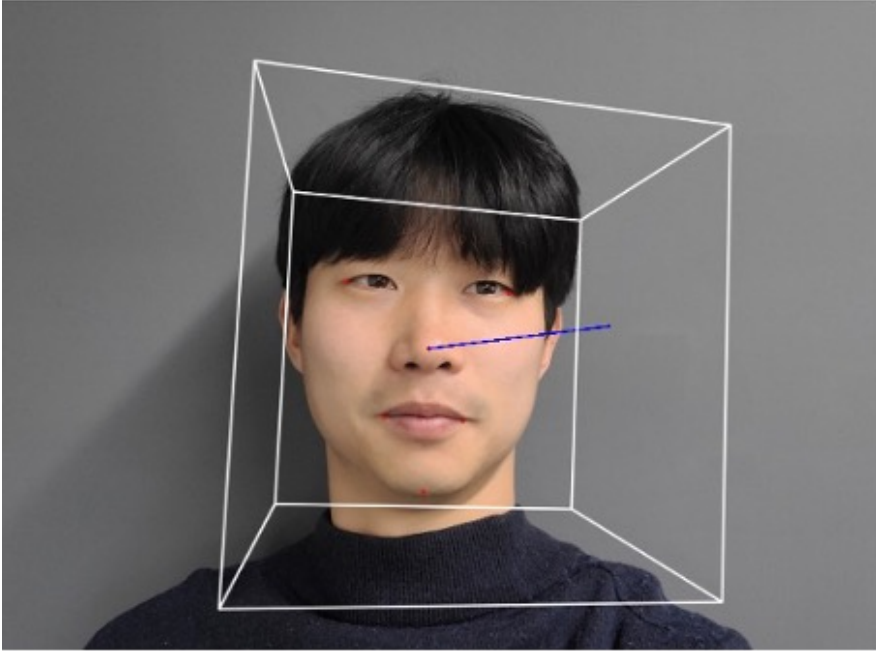


그림 17: Facial landmarks와 head pose 예측

3.4 눈 크기 교정

앞에서 머리 자세 추정으로 구했던 Rotation의 값들중에 [그림.18]에 있는 YAW 의 값을 이용 해서 Eye Aspect Ratio (EAR)을 교정하는 [수식.(??)]을 사용 하여 계산하였다. 이를 통해서 운전자가 측면을 바라 볼때 눈의 크기의 비율이 달라 지는 것을 교정하였다.

$$CorrectEAR = EAR - \frac{\|Yaw\| - \epsilon}{100} \quad (3.1)$$

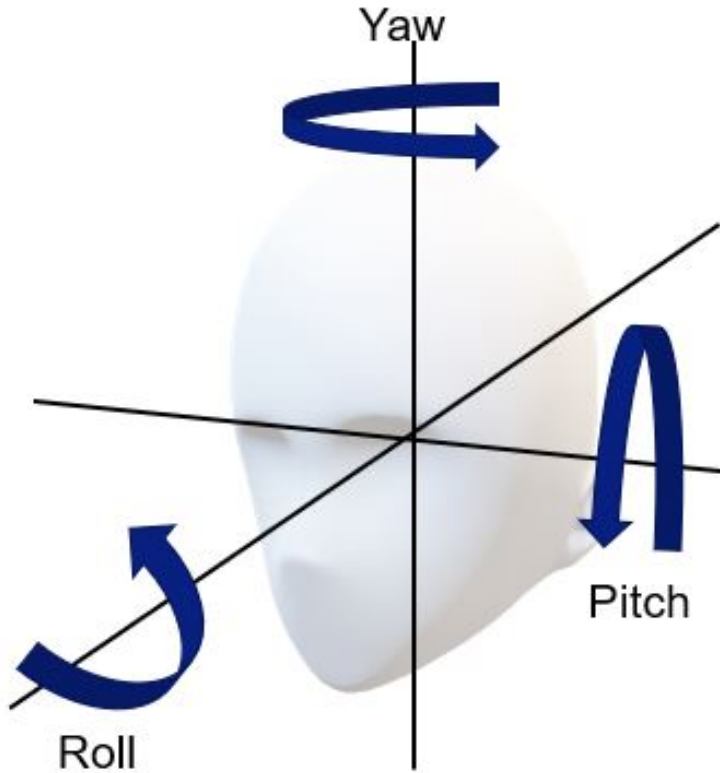


그림 18: 머리 자세에서 구한 Yaw, Roll, Pitch

3.5 눈깜빡임 감지

눈 깜빡임을 감지하기 위해서 EAR 이라는 척도를 사용하였다. EAR 은 앞에서 얼굴을 감지하고, 얼굴 특징점을 찾아서 눈의 특징점의 6개의 지점을 찾아서 가로와 세로의 비율을 계산하는 하기의 수식(3.2)을 사용하여 계산한다. 눈의 단순 크기만을 측정하게 되면 운전자의 의자가 멀리 있는 자율운전 주행환경에서 눈의 크기가 작게 계산되는 오류를 범할수 있기 때문에 눈의 가로 세로 비율을 통해서 동작하도록 구현하였다. 추가

적으로 사람마다의 EAR이 다를수 있으므로, 본 논문에서는 주행 최초의 5초동안의 평균 EAR을 계산하여 평균 EAR을 Threshold로 잡도록 구현 하였다. 그리고 해당 시스템은 실시간 졸음 감지를 해야하기 때문에 눈이 한번 깜빡일때 [그림20]과 같이 150ms에서 200ms 정도 소요 되는점을 고려한다면 최소한의 FPS는 10fps 가 필요하고, 조금더 정밀하고 최근 카메라의 성능이 최소한 30fps는 지원하기 때문에 30fps의 stream을 처리 하도록 구현하였다.

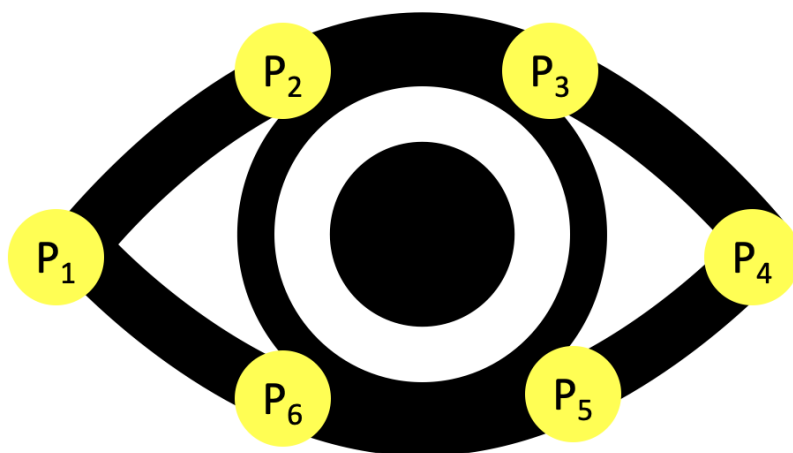


그림 19: EAR을 위한 Point

$$EAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|} \quad (3.2)$$

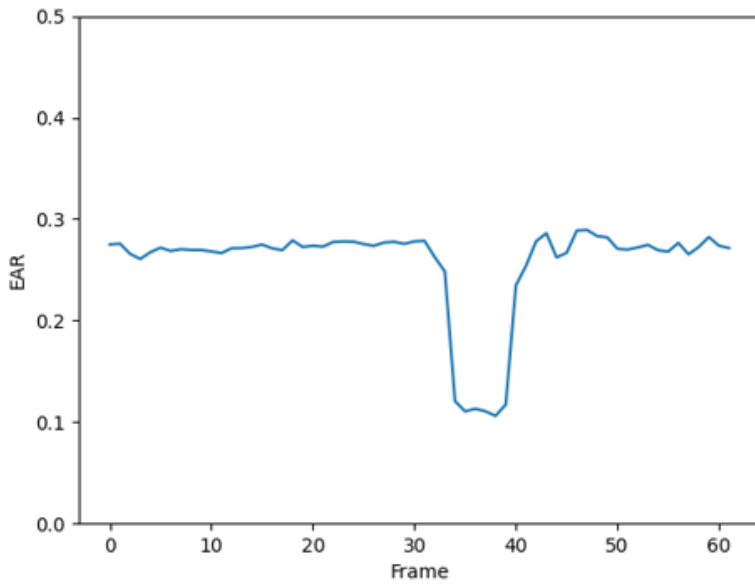


그림 20: Frame 에 따른 EAR 변화

3.6 하품 감지

하품을 감지하기 위해서 사용한 척도는 Mouth Aspect Ratio (MAR)이다. MAR도 앞서 설명한것처럼 얼굴 주요 특징점을 찾아서 [그림21]과 같이 8개의 포인터를 찾고, 가로와 세로의 비율을 수식(3.3)과 같이 사용한다. L. Li[18] 에서 설명한것처럼 MAR이 0.5 보다 크고 5초 이상 지속 되면 하품으로 판단하였다.

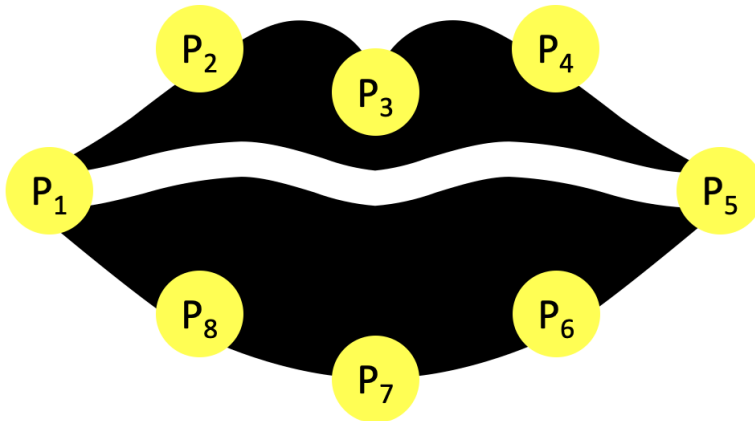


그림 21: MAR을 위한 Point

$$MAR = \frac{\| p2 - p8 \| + \| p3 - p7 \| + \| p4 - p6 \|}{3 \| p1 - p5 \|} \quad (3.3)$$

3.7 졸음 감지를 위한 변수들

졸음을 감지하는 여러가지 변수들이 존재 한다. 본 논문에서는 눈 깜빡임 지속시간, 눈 깜빡임 빈도수, PERCLOS, 운전 지속 시간을 측정하여 졸음 감지의 단계를 구분하였다.

- 눈깜빡임 지속시간(ms): 일반적으로 눈의 깜빡임 지속 시간은 J.L. Andreassi[19]가 설명한것처럼 200-400ms입니다. 그리고 A. Chowdhury[20] 의 논문에서처럼 눈이 감은상태로 0.5초 이상 지속되면 이것은 micro-sleep으로 간주하였다.
- 눈깜빡임 빈도수(blinks/min): 눈의 깜빡임은 일반적으로 평온한 상태에서 1분동안 15회에서 20회정도로 깜빡이지만, 졸린 상태에서는 20회 이상 또는 5회 미만으로 깜빡인다는 U. Svensson [21] 의 내용으로 졸음을 간주하였다.
- PERCLOS: 1분동안 눈을 80% 이상 감고 있는 시간의 비율도 졸음의 지표가 될 수 있다[20].
- 운전지속시간: 운전하는 시간이 18분이후 부터 1시간사이부터는 졸음을 할 가능성이 커진다[2].

제 4 장

실험 결과

4.1 실험환경

해당 실험을 하기 위해서 자율주행이 되는 자동차에서 장기간 운전 을 하면서 실험을 해야한다. 처음에는 모형실험으로 2.3GHz 쿼드 코어 Intel Core i5의 Mac Book Pro에서 환경으로 시작하였다. 모형실험을 할때 에 실험자들이 앞에 화면이 없고, 카메라만 있는 상태로 실험을 하다보니 실험에 집중을 하지 못하고 눈의 시선이 분산되는 경우가 많아서 Bench Test를 [그림.22]과 같이 진행하였다. 실험의 구현은 Python과 OpenCV와 Tensorflow를 주로 사용하여 구현을 하였다. 실험 장비로는 MacBook Pro (8G memory, 2.3GHz Intel Core i5) 를 사용해서 자동차 임베디드 시스 템의 사양과 비슷하게 구성하였다. 카메라는 30fps, 1280 x 720 해상도를 가진 Intel real sense d435 제품을 사용하였다. 해당 실험을 위해서 10명의 지원자를 구해서 실험을 진행하였다.



그림 22: Bench Test 환경

4.2 얼굴 감지 모델 성능 비교

앞서 설명한 본 연구의 구성도에서 알수 있듯이 실험을 진행하기 위해서 여러가지 과정이 필요하다. 처음으로는 우선 얼굴을 감지해야한다. 얼굴을 감지하는 방법에 대해서는 정면얼굴을 잘 감지하는것으로 알려져있는 머신러닝 기법으로 HoG Feature 와 SVM을 기반으로 한 모델을 통해서 얼굴을 감지하는 방법을 먼저 실험하였다. 그리고 딥러닝 기법의 MobileNet-SSDv2을 사용해서 얼굴을 감지하는 방법을 실험하였다.

실험은 해상도에 따른 수행 시간을 비교를 하여 고해상도 카메라에서도 실시간으로 성능을 보장할수 있는지 확인하였다. HOG & SVM 방법으로 얼굴 감지 하였을때에는 [표.5]와 같이 해상도가 올라가면 올라갈수록 FPS가 10FPS를 넘지 못하면서 실시간 감지 시스템이라고 말할수 없게 되었다. 하지만 MobileNet SSD v2를 사용해서 얼굴 감지를 했을때에는 [표.6]와 같이 해상도가 올라가더라도 수행 시간은 거의 변화지 않는것을 알수 있었다.

그리고 각 모델의 Average IOU와 mAP를 비교하는 실험을 하였다. HOG&SVM 모델의 Average IOU는 0.253이고 mAP는 0.365이지만 MobileNet SSD v2 모델은 Average IOU가 0.598이고 mAP는 0.751로 상당한 성능 차이를 가지고 있다.

표 5: HOG&SVM 해상도에 따른 수행 시간

해상도	수행시간	FPS
320x240	35 ms	28 fps
720x480(SD)	140 ms	7 fps
1280x720(HD)	420 ms	2 fps
1920x1080(FHD)	980 ms	1 fps
3840x2160(UHD)	3600 ms	0.2 fps

표 6: MobileNet SSD v2 해상도에 따른 수행 시간

해상도	수행시간	FPS
320x240	26.6 ms	37 fps
720x480(SD)	26.5 ms	37 fps
1280x720(HD)	27.5 ms	36 fps
1920x1080(FHD)	26.6 ms	37 fps
3840x2160(UHD)	27.2 ms	36 fps

4.3 뜬눈의 EAR과 감은눈의 EAR 차이 실험

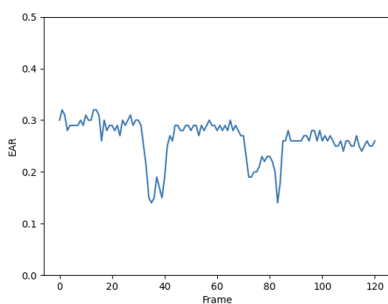
눈의 깜빡임을 감지하기 위해서 10명의 지원자의 EAR을 계산하고 눈을 떴을때의 평균과 눈을 감았을때 최저점을 구해서 두 사이의 차이값을 구하여 보았다. [표.7]와 같이 10명의 지원자들의 뜬눈의 평균과 감은눈의 최저점의 차이는 모두 다른것을 알 수 있었다. 따라서, Threshold 값을 정하는것에 특정한 값을 넣는것 보다는 운전 시작 5초의 평균값을 항상 계산하여 Threshold 값을 동적으로 정하였다.

표 7: 뜬눈의 EAR값고 감은 눈의 EAR 실험

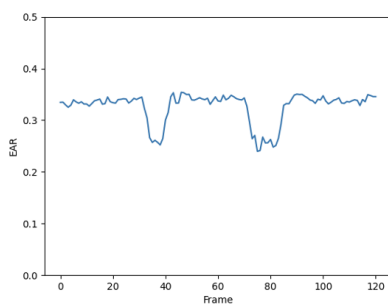
지원자	EAR 뜬눈 평균	EAR 감은눈 최저점	차이
1	0.26	0.12	0.14
2	0.32	0.11	0.21
3	0.33	0.20	0.13
4	0.38	0.18	0.20
5	0.45	0.21	0.24
6	0.23	0.12	0.11
7	0.42	0.19	0.23
8	0.28	0.13	0.15
9	0.40	0.21	0.19
10	0.32	0.14	0.18

4.4 EAR 감지기 성능 비교 실험

본 실험에서는 [15] 논문에서 실험했던 HOG & SVM 모델과 Cascade Regression Function을 이용한 방법과 본 연구에서 제안한 MobileNet SSD v2로 얼굴을 감지하고 Cascade Regression Function을 통해서 EAR을 도출한것을 비교하는 실험을 진행하였다. [그림.23]과 같이 HOG&SVM 모델과 Cascade Regression 모델을 사용했을때 EAR의 결과와 본 논문에서 제안한 방법으로 EAR을 추출한 결과는 정면에서는 크게 다르지 않은것을 볼수 있다. 하지만 측면으로 보고 있을때에는 [그림.24]과 같이 큰 차이의 성능을 가지는것을 확인할 수 있었다. 그 이유는 얼굴을 옆으로 돌리게 되었을때 EAR을 계산하게되면 가로와 세로의 비율은 같은 눈의 크기라도 달라지기 때문인것으로 판단하였다. 이러한 결과를 가지고 실제적으로 시스템에서 비교했을때 [그림25과 같이 알고리즘을 적용전에는 EAR 수치가 평균치보다 큰 0.51 정도로 나오는 오탐지를 하고 있다. 하지만, 보정 알고리즘을 적용하게 되면 EAR 수치가 평균치인 결과로 0.39정도로 나오는것을 확인 할수 있었다.

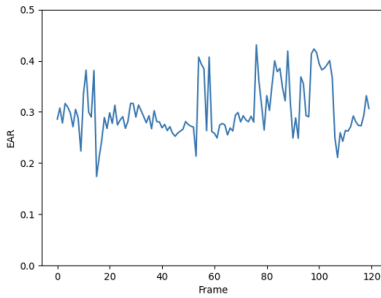


(a) HOG & SVM 과 Cascade Regression 모델

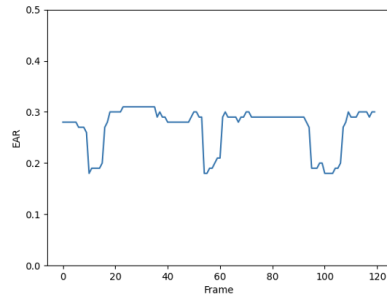


(b) MobileNet-SSD v2 와 Cascade Regression 모델

그림 23: 정면에서 EAR 성능 비교



(a) HOG & SVM 과 Cascade Regression 모델



(b) MobileNet-SSD v2 와 Cascade Regression 모델

그림 24: 측면에서 EAR 성능 비교

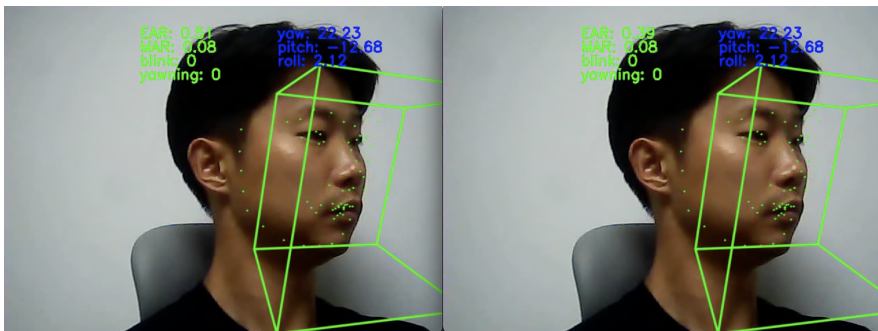


그림 25: 머리자세에 따른 EAR보정 결과

4.5 눈 깜빡임 성능 실험

다음으로는 눈 깜빡임 성능에 대해서 얼굴의 정면과 측면에 대해서 10명의 지원자에서 5분동안 추출한 비디오 데이터를 가지고 오차행렬을 통해서 Precision, Recall과 F1 Score를 실험한 결과를 하기의 [표.8]에 나타내었다. 평균적으로 얼굴 정면의 Precision은 0.99 였고, Recall은 0.97, F1 Score는 0.98로 나와서 우수한 성능을 가진것을 확인할 수 있었다. 추가적으로 얼굴의 측면에서도 [표.9]와 같이 평균적으로 Precision이 0.96, Recall이 0.95, F1 Score가 0.96으로 우수한 결과를 확인할 수 있었다.

표 8: 얼굴 정면 눈 깜빡임 성능

지원자	TP	TN	FP	FN	Precision	Recall	F1 Score
1	92	5	0	4	1.00	0.96	0.98
2	82	2	1	2	0.99	0.98	0.98
3	86	4	2	4	0.98	0.96	0.97
4	76	0	1	1	0.99	0.99	0.99
5	67	2	1	0	0.99	1.00	0.99
6	79	1	0	1	1.00	0.99	0.99
7	89	2	1	9	0.99	0.91	0.95
8	92	1	0	3	1.00	0.97	0.98
9	68	3	3	5	0.96	0.93	0.94
10	67	3	2	1	0.97	0.99	0.98
평균	79.8	2.3	1.1	3	0.99	0.97	0.98
표준편차	10.04	1.49	0.99	2.67	0.01	0.03	0.02

표 9: 얼굴 측면 눈 깜빡임 성능

지원자	TP	TN	FP	FN	Precision	Recall	F1 Score
1	83	2	3	4	0.97	0.95	0.96
2	77	4	3	4	0.96	0.95	0.96
3	84	3	5	5	0.94	0.94	0.94
4	68	6	3	1	0.96	0.99	0.97
5	74	3	6	5	0.93	0.94	0.93
6	69	4	2	5	0.97	0.93	0.95
7	82	4	3	2	0.96	0.98	0.97
8	62	1	3	4	0.95	0.94	0.95
9	67	2	3	4	0.96	0.94	0.95
10	87	3	1	3	0.99	0.97	0.98
평균	75.3	3.2	3.2	3.7	0.96	0.95	0.96
표준편차	8.56	1.40	1.40	1.34	0.02	0.02	0.01

제 5 장

결론

5.1 연구 결과

본 연구에서는 자율 주행 환경에서 운전자가 카메라의 사이의 거리와 상관없이 눈의 가로 세로의 비율을 통해서 눈깜빡임이 있었는지 판단하고 운전자가 어떤 물체를 주시하기 위해서 아래를 보거나 옆을 보더라도 졸음을 제대로 판단할수 있도록 EAR을 보정하고 눈 깜빡임을 인식할수 있는 시스템을 구현하고 검증하였다. 그리고 얼굴 정면의 눈 깜빡임은 Precision 0.99, Recall 0.97, F1 Score가 0.98로 G. Salzillo[15] 보다 각각 0, 0.01, 0.02 우수한 성능을 보였다. 측면에서는 눈 깜빡임에 대해서 Precision 0.96, Recall 0.95, F1 Score 0.96으로 G. Salzillo[15] 보다 각각 0.04, 0.03, 0.04 우수한 성능을 보이는 것을 확인할 수 있었다. 또한, 자동차안에 사용하는 카메라의 기능이 복합적으로 사용해지면서 높아지는 해상도에서도 실시간으로 동작할 수 있는 시스템을 본 연구에서 설계하였다. 본 연구를 통해서 현재 시중에 팔리고 있는 자동차들의 졸음 방지 시스템이 자율주행 환경에서 더 나은 졸음 감지를 할수 있도록 방법을 제시하였다. 앞으로 완전 자율 주행 단계에 이르기까지 오랜 시간이 필요할 것으로 예상되고 있는 가운데 본 논문은 현재 2.5단계 자율 주행 단계에서 운전자들의 안전한 운전을 도움을 줄것으로 예상된다.

5.2 후속 연구 과제

본 연구를 통해 자율주행 환경에서 운전자들이 졸음운전을 했을 때 어떻게 졸음을 감지하는지에 대한 토대를 마련하고 자율 주행 2.5단계 환경에서 운전자 졸음 감지에 중요한 점이 무엇인지 제시 하였다. 본 연구에서 취득한 데이터셋을 통해서 다양한 전처리와 후처리를 추가하면서 어떤 추가적인 인자들이 졸음의 레벨을 올려주는지 확인할수 있는지 후속 연구가 이루어 질 필요가 있다. 또한, 카메라를 이용한 방법 이외에도 자동차에 마이크가 탑재 되어 있는 사실을 기반으로 음성에 대한 요소를 추가적으로 연구하여 운전자가 말하지 않고 조용할때와 차량의 음악소리가 나오지 않을때를 판단하여 졸음 레벨을 올려주는 연구를 진행 해야할것이다. 본 연구에서는 RGB 카메라만을 사용하여 운전자의 졸음을 판단하고 있지만, RGB 카메라와 IR 카메라를 동시에 사용해서 연구를 진행한다면 조금더 신뢰성이 있는 시스템을 만들수 있을것이다.

참고 문헌

- [1] H. Ueno, M. Kaneda, and M. Tsukino, “Development of drowsiness detection system,” in *Proceedings of VNIS'94 - 1994 Vehicle Navigation and Information Systems Conference*, pp. 15–20, 1994.
- [2] T. Arakawa, “Trends and future prospects of the drowsiness detection and estimation technology,” *Sensors*, vol. 21, no. 23, 2021.
- [3] M. S. Doudou, A. Bouabdallah, and V. Cherfaoui, “A Light on Physiological Sensors for Efficient Driver Drowsiness Detection System,” *Sensors & Transducers Journal*, vol. 224, pp. 39–50, Aug. 2018.
- [4] C. Morimoto, D. Koons, A. Amir, and M. Flickner, “Pupil detection and tracking using multiple light sources,” *Image and Vision Computing*, vol. 18, no. 4, pp. 331–335, 2000.
- [5] Z. Zhu, K. Fujimura, and Q. Ji, “Real-time eye detection and tracking under various light conditions,” in *Proceedings of the 2002 Symposium on Eye Tracking Research and Applications*, ETRA '02, (New York, NY, USA), p. 139–144, Association for Computing Machinery, 2002.
- [6] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, 2001.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural*

Information Processing Systems (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

- [9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, pp. 1499–1503, oct 2016.
- [10] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen, and T.-T. Lee, “Mobilenet-ssdv2: An improved object detection model for embedded systems,” in *2020 International Conference on System Science and Engineering (ICSSE)*, pp. 1–5, 2020.
- [11] Y. Zhang and C. Hua, “Driver fatigue recognition based on facial expression analysis using local binary patterns,” *Optik*, vol. 126, no. 23, pp. 4501–4505, 2015.
- [12] C. S. C. A. e. a. Picot, A., “Using retina modelling to characterize blinking: comparison between eog and video analysis,” *Machine Vision and Applications*, vol. 23, pp. 1195–1208, 2012.
- [13] M. W. Akrouf, B., “Spatio-temporal features for the automatic control of driver drowsiness state and lack of concentration.,” vol. 26, p. 1–13, 2015.
- [14] W. Deng and R. Wu, “Real-time driver-drowsiness detection system using facial features,” *IEEE Access*, vol. 7, pp. 118727–118738, 2019.
- [15] G. Salzillo, C. Natale, G. B. Fioccola, and E. Landolfi, “Evaluation of driver drowsiness based on real-time face analysis,” in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 328–335, 2020.
- [16] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

- [17] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5525–5533, 2016.
- [18] L. Li, Y. Chen, and Z. Li, “Yawning detection for monitoring driver fatigue based on two cameras,” in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 1–6, 2009.
- [19] J. L. Andreassi, “Psychophysiology: Human behavior and physiological response,” vol. 65, pp. 174–175, 2007.
- [20] A. Chowdhury, R. Shankaran, M. Kavakli, and M. M. Haque, “Sensor applications and physiological features in drivers’ drowsiness detection: A review,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3055–3067, 2018.
- [21] U. Svensson, “Blink behaviour based drowsiness detection: method development and validation,” 2004.

Abstract

Real-time Eye Blink Detection System for Driver's Side Face in Autonomous Driving Environment

Bonhak Ku

Graduate School of Practical Engineering

Seoul National University

As many cars in the second level of autonomous driving (SAE level 2) start to appear, drivers are shifting their gaze to the car's entertainment system instead of focusing on driving. However, there are still several steps left ahead until we reach the stage of fully autonomous driving. Meanwhile, we need a system that can help drivers focus on the road and avoid drowsiness.

Previous studies focused on methods for detecting drowsiness have been conducted in a manual driving environment when the driver is looking straight ahead, not in an autonomous driving environment when driver is looking in different directions. Therefore, this study researched and developed a drowsiness prevention system in an autonomous driving environment by a non-contact method, using a camera. The system warns drivers to stay focused on driving when they are paying attention to the entertainment system or feeling drowsy.

The drowsiness of the driver was determined by the following steps. First, the driver's face was recognized, followed by the main points of the driver's face. Then, the necessary main points of the eye and mouth were found. Finally, the pose of the head was determined through the found main points. Additionally, the system was designed with an algorithm that corrects the eye size when the head is turned to the side, a task many systems in the market fail to achieve.

In this paper, OPENCV was mainly used in the python development environment. High-performance MobileNet-SSD is adopted as a face detection model. For facial landmark detection, regression trees were used. As for the head posture, the Perspective-n-Point method using the 2D coordinates of the face points found earlier, were used to estimate the direction of the face rotation, and also correct the size of the eye from the side view of face.

Through this study, it was confirmed that in a level 2 autonomous driving environment, the driver's drowsiness can be better detected when the driver is looking to the side. Applying the results of this study on a real word product will help determine the driver's drowsiness more accurately, thus securing software competitiveness.

Keywords : Real-time, autonomous driving, drowsiness detection

Student Number : 2021-26571

찾아보기

caption, 15, 26, 35, 36, 38

27, 29, 31, 33, 37

equation, 27, 29, 31

table, 15, 26, 35, 36, 38

figure, 1-4, 7, 8, 12, 16, 22, 25-

tabular, 15, 26, 35, 36, 38