경운대학교 KYUNGWOON UNIVERSITY

경운대학교 소프트웨어중심대학사업단

[서식 5] CLP 학기 최종 결과보고서

CLP 학기 최종 결과 보고서

과 제 명	ChatGPT와 영상처리를 이용한 졸음 감지 시스템				
팀 명	Clean Code				
학 과 명	소프트웨어학부			지도교수	오 세 진
참여기업	기업명	(주)컴퓨터메이트		전화	053-812-3008
	대표자	서강인, 김성호		담장자	김성수(010-2943-0280)

구 분		성 명	학번	학년	휴대전화	e-mail
참여 학생	팀장	이현준	201805013	3	010-4940-9501	bmjjl2@naver.com
	팀원 *필요 시 확장	순현상	201905013	3	010-4819-7811	tnsgustkd@naver.com
		조성훈	201905057	3	010-4132-0445	mc213213@naver.com
		서창희	201905031	3	010-6644-8907	wldnjs3608@naver.com
		강지윤	202105011	3	010-5165-5658	wldbs5165@naver.com

4차 산업혁명 혁신선도대학 사업의 CLP교과운영에 따른 최종 결과 보고서를 제출합니다.

첨부 1. 최종 결과 보고서 1부.
2. 학생 만족도 조사 1부. 끝.

2023. . .

대 표 학 생 : 이현준 (인)
팀 지도교수 : 오세진 (인)

경운대학교 소프트웨어중심대학사업단장 귀하

[서식 5-1] 최종 결과 보고서

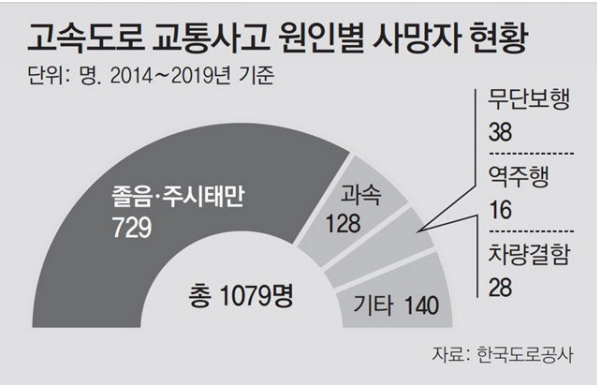
최종 결과 보고서

1. 과제 개요

가. 과제 선정의 배경

졸음 및 주시태만이 고속도로 교통사고의 가장 큰 원인으로 나타났다. 한국도로공사에 따르면 2014~2019년 도로공사가 관리하는 고속도로에서 발생한 교통사고 사망자는 총 1079명이었는데, 그중 졸음 및 주시 태만으로 인한 사망자가 729명으로 전체의 67.6%를 차지하고 있다. 더욱이, 경찰청 통계(2017년)에 따르면 졸음운전으로 인한 교통사고 사망률은 4.51%로 음주운전 사망률보다 1.75배 높은 것으로 나타났다. 졸음운전은 혈중 알코올농도 0.17%의 만취 상태로 운전하는 것과 유사한 위험이 존재한다.

졸음운전 사고의 대다수인 82.5%가 운전자 혼자 탔을 때 발생하는데 그 이유는 운전자 주의를 환기해줄 사람 즉, 대화할 사람이 없어서 발생한다고 한다. 실제로 졸음운전은 장시간운전, CO2 농도 증가와 같은 요인으로 집중력 저하에서 발생하는데 찾아본 논문에 의하면 대화는 뇌를 활성화해주는데 뇌의 활성화는 집중력을 향상해 준다고 한다. 따라서 우리는 제한된 환경에서도 동작이 가능한 임베디드 보드에 영상처리와 Chat GPT를 이용하여, 차량 내에서 운전자의 졸음을 판단하고 대화와 퀴즈를 통해 졸음을 깨워주는 졸음감지 시스템을 개발을 목표로 선정하였다.



나. 과제 개발 혹은 제작에 따른 기대효과

차라는 제한된 환경에서 임베디드 보드를 이용하여 영상처리로 졸음을 감지하고 경고음 과 ChatGPT를 이용한 운전자와의 소통을 통해 운전자의 주의력을 빠르게 회복시켜주도록 시스템을 제공해 줌으로써 졸음운전에서 벗어나 안전운전을 할 수 있을 것이라 기대된다. 그리고 임베디드 보드에서 원활한 동작을 위해 코드 경량화나 최적화가 성능향상에 얼마만큼 영향을 주는지 또 코드를 동작하면 어느 임베디드 시스템의 성능이 가장 좋을지 등의 비교를 해가면서 PC 이외에 임베디드 환경에서도 원활한 동작이 가능 할 것이라 기대한다. 그 외에도 ChatGPT를 이용한 소통기술은 다양한 기기들에 이용하면 사용자에게 편의를 제공해 줄 수 있고 특정 텍스트를 인식해 사용자에게 자동화된 환경도 제공 가능할 것이라 기대된다. 목표 시스템을 구축 추후에 Co2 농도 센서를 이용해 차 내부에 Co2농도를 알아 운전자에게 "Co2농도가 높습니다. 창문을 열어주세요."와 같은 여러 가지 도움을 주는 시스템을 추가해 나가면 운전자에게 보다 더 졸음운전에서 벗어날 수 있게 제공하는 시스템이 구축될 것이라고 기대된다.

2. 활용 / 관련 이론

구 분	관련 이론 및 지식
전공 분야	OpenCV와 Dlib 영상인식을 활용한 졸음감지, Chat GPT를 활용한 운전자와의 소통, 임베디드 개발보드 이용
비 전공 분야	Perclos 공식, EAR(eye aspect ratio) 알고리즘

3. 역할분담

연번	성명	담당	수행 역할
1	이현준	팀장 및 총괄. 동작 코드 개발	팀 총괄, 전체적인 프로그램 기획 및 개발, 동작 환경구축. 전체적인 코드 개발 및 코드 경량화, 최적화
2	순현상	Jetson Nano 환경구축, 각 환경 동작 성능 비교	Jetson Nano 환경의 동작을 위한 OS, 라이브러리 호환성 검사 및 설치, 각 동작환경의 동작 Fps 출력 코드 추가 및 성능 비교
3	조성훈	Raspberry Pi 환경구축, 영상처리 코드 개발	Raspberry Pi 환경의 동작을 위한 OS, 라이브러리 호환성 검사 및 설치, 눈의 뜨고 감음을 판별하는 코드 개발
4	서창희	Raspberry Pi 4 환경구축, 졸음감지 코드 개발	Raspberry Pi 환경의 동작을 위한 OS, 라이브러리 호환성 검사 및 설치, 졸음감지를 판별하는 공식의 코드 개발
5	강지윤	Raspberry Pi 4 환경구축, ChatGPT소통 코드 개발	Raspberry Pi 환경의 동작을 위한 OS, 라이브러리 호환성 검사 및 설치, ChatGPT와 운전자와의 소통을 위한 라이브러리, 음성파일 추가 및 동작 코드 개발

4. 과제의 수행 정도

연번	추진 내용	1주	2주	3주	4주	5주
1	기획 및 설계	■				
2	개발 환경 구축	■	■			
3	얼굴 인식 및 졸음 식별 코드 설계 및 구현		■	■		
4	ChatGPT를 이용 소통 하는 코드 설계 및 구현		■	■		
5	코드 개선			■	■	
6	임베디드 개발보드 구동 테스트		■	■	■	
7	테스트 및 보완				■	■

5. 세부 구성내용

1) 서 론

1-1. 배경 및 목적

○ 영상처리를 이용한 졸음감지와 ChatGPT를 이용한 운전자와의 소통을 통해 운전자가 다시 졸음운전을 하지 않게 감지 및 도움을 주는 시스템을 개발하게 되었다. 또 자동차 환경자체가 제한된 환경이여서 PC말고 임베디드 보드를 이용해 자동차에서도 동작이 가능하게 하였다. 그리고 각 환경의 동작 성능을 비교도 해보고 임베디드 환경의 원활한 동작을 할 수 있게 다양한 방법을 찾아 운전자에게 임베디드 보드로도 보다 원활한 시스템을 제공할 수 있도록 개발 방향을 선정하였다.

2) 본 론

2-1. 시스템 환경

우린 최종적으로 임베디드 보드에서의 동작의 목표여서 원활한 코드 개발이 가능한 PC환경을 호스트 시스템으로 이용하고 그 후 임베디드 보드 즉 타켓 시스템에 코드를 적용하여 동작하게 개발 진행하였다. 본 시스템에 개발에 사용된 시스템 사양과 타겟 시스템의 사양, 사용된 라이브러리는 아래 표1 ~ 표4와 같다.

구분	버전
Scipy	1.11.4
Imutils	0.5.4
Time	
Dlib	19.17.0
Cv2	4.5.1
Pygame	1.9.5
Pyttsx3	
Os	
Openai	0.28.0
Dotenv	
Speech_recognition	3.10.0
gTTS	2.4.0
Thread	

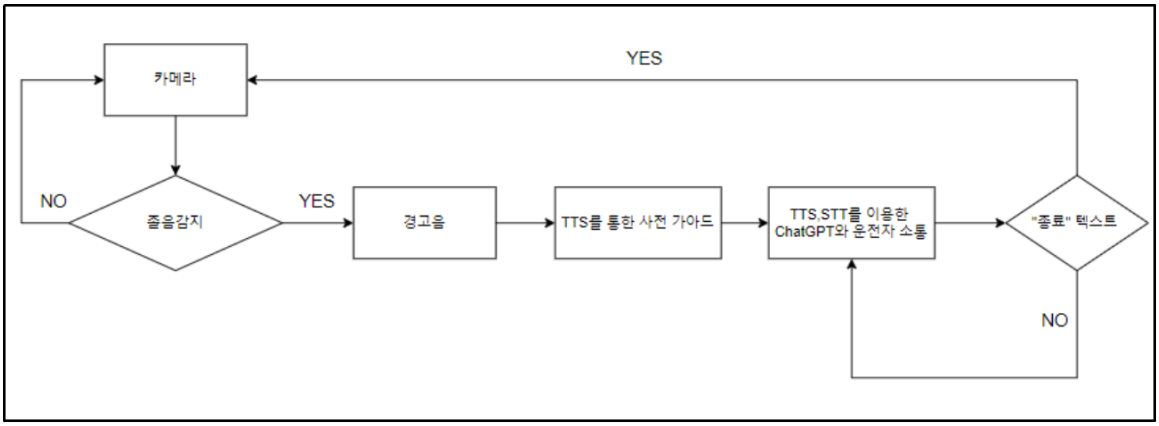
구분	사양
CPU	Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz
RAM	8.0GB
OS	Window 11
시스템 종류	64bit
GPU	NVDIA GeForce GTX 750

구분	사양
CPU	APM Cortex-A72 1.5GHz
RAM	8GB
OS	Raspberry Pi OS Legacy
시스템 종류	64bit
SD	32GB
GPU	ARM Cortex-A72 1.5GHz

구분	사양
CPU	Quad-core ARM A57 @ 1.43 GHz
RAM	4GB
OS	Ubuntu 18.04
시스템 종류	64bit
SD	32GB
GPU	128-core Maxwell

2-2. 시스템 설계

임베디드 보드에 연결된 카메라를 통해 Dlib, Opencv, Scipy 라이브러리를 이용해 운전자의 졸음을 감지한다. Perclos 공식을 이용해 운전자가 졸음운전을 하고 있다고 감지하면 연결된 스피커를 통해 경고음을 발생하고 운전자와의 소통이 광범위해지지 않게 사전 가이드를 TTS를 통해 음성으로 운전자에게 전달해 준다. 그에 대한 운전자의 응답을 STT를 통해 ChatGPT에 전달하고 생성된 응답을 TTS를 통해 운전자에게 전달해 주는 식으로 소통이 이루어진다. 운전자의 응답을 STT 했을 때 "종료"라는 텍스트가 있으면 소통 부분이 종료되게 된다.



2-3. 활동 사진

매주 팀 회의를 통해 주마다 우선순위를 정해 인원 분배 하여 보다 효율적으로 개발 진행하였다. 코드 동작 확인을 순차적으로 해 가면서 오류를 바로 확인하고 수정 할 수 있게 개발 하였다. 404호에 개발 공간을 만들어 매주 활동 하였으며 Slack를 구축하여 팀원들과 코드나 자료를 공유하였다.

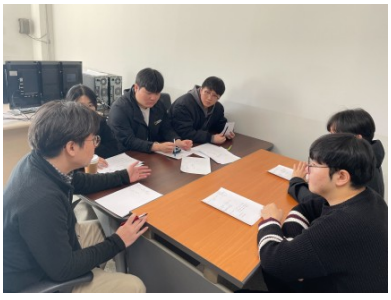


그림 6 팀 회의

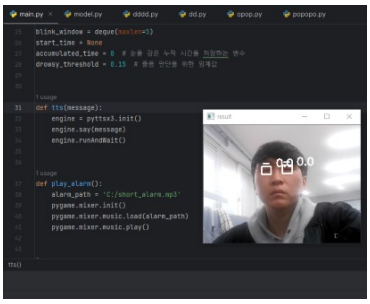


그림 7 동작확인

그림 8 팀 활동

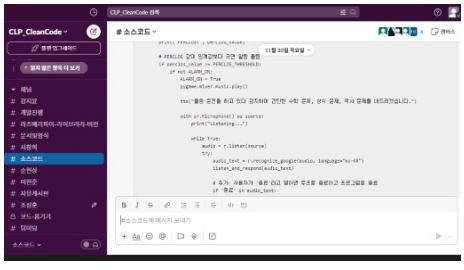


그림 9 Slack 구축

2-4. 시스템 구현

2-4-1. 눈 감은 유무 판별

처음엔 라벨링된 데이터를 이용해 Pytorch를 이용해 학습하여 눈 감은 유무를 판별 하였는데 눈이 작은 사람은 눈 을 떠도 감았다고 판별하는 문제가 빈번하게 발생을 하여 Soukupová와 Čech의 2016년 논문인 Facial Landmarks 를 사용한 Real-Time Eye Blink Detection 의 작업을 기반으로 EAR 관계를 반영하는 방정식을 이용하여 Dlib와 Scipy라이브러리를 이용해 눈의 운전자의 눈의 종횡비를 구해 판별하는 식으로 수정해 개선하였다.

그림 10 데이터 학습 통한 판별

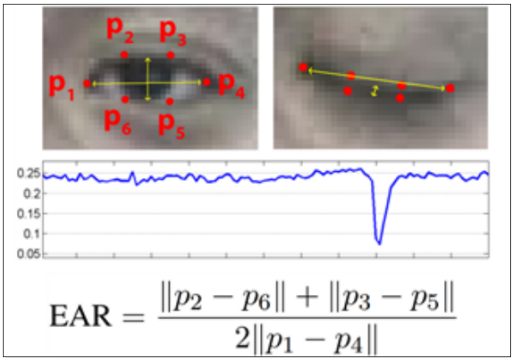


그림 11 눈의 종횡비를 구해 판별

그림 12와 그림 13을 비교해보면 확실하게 눈의 종황비를 구해 판별하는 쪽이 눈을 뜨고 있다고 판별하여 1을 출력하고 있는걸 확인 할 수 있다.

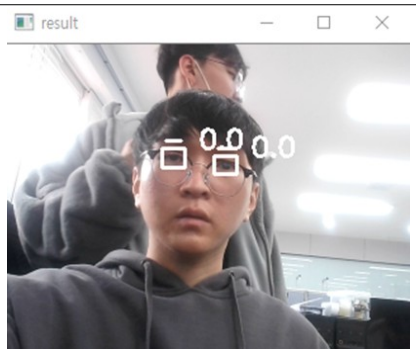


그림 12 데이터 학습을 통한 판별

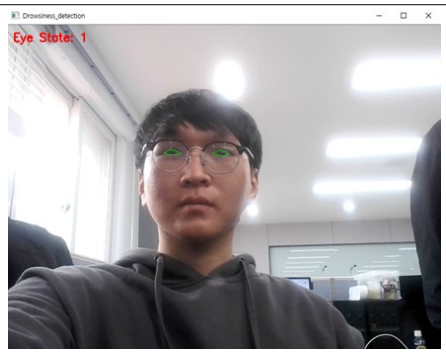


그림 13 눈의 종횡비를 구해 판별

2-4-2. 졸음감지

운전자의 졸음감지는 PERCLOS공식을 이용했다. 누적을 위한측정 시간을 선언을 하고 측정시간 동안 운잔자의 눈 감은 시간을 누적하여 공식에 적용한 뒤에 값이 0.15이상이면 운전자가 졸음운전을 하고 있다 판별하게 설계하였다.

$$PERCLOS = \frac{\text{눈 감은 시간의 누적}}{\text{누적을 위한 측정 시간}}$$

State	PERCLOS value
Awake	0.0 ~ 0.074
Questionable	0.075 ~ 0.149
Drowsy	0.15 ~

그림 14 PERCLOS 공식 및 판단 기준

2-4-3. 운전자와 ChatGPT 간 소통

Openai, Dotenv 라이브러리 등을 이용하여 ChatGPT API를 발급받아 코드에 연동하여 구현하였다. 운전자의 소통이 이루어지기 전에 스피커를 통해 경고음이 먼저 발생하도록 구현하였다.

```
# TTS 및 ChatGPT 소통
tts("졸음 운전을 하고 있다 감지하여 간단한 수학 문제, 상식 문제, 역사 문제를 내드리겠습니다.")
with sr.Microphone() as source:
    print("Listening...")
```

GPT-3.5-turbo를 사용하였고 사전 정의된 텍스트를 TTS를 통해 운전자에게 음성으로 전달해 줌으로써 운전자와의 대화가 광범위해지지 않도록 설계하였다. 문제를 내달라고 요청할시 정답도 같이 말하는 문제가 있어 ? 뒤에 정답이 나온다는 공통된 특징을 찾아 Split을 이용해 ?뒤에 오는 텍스트는 제거하여 문제를 해결하였다.

```
You said: 간단한 수학 문제 내 줘
간단한 수학 문제 내 줘
7 + 3
generating audio
speaking
speaking complete
PERCLOS: 0.1038961038961039
PERCLOS: 0.18142548596112312
result2:
{   'alternative': [{'confidence': 0.87048674, 'transcript': '정답은 10'}],
    'final': True}
You said: 정답은 10
정답은 10
맞았습니다!
```

그림 16은 사전 정의된 음성을 TTS로 운전자에게 전달한 뒤에 운전자가 문제를 내달라고 요청한 음성을 STT를 통해 ChatGPT에 전달되고 생성된 텍스트를 TTS를 통해 운전자에게 전달해주는 과정이 실행되고 있는 사진이다. 종료는 운전자의 응답을 텍스트 형식으로 바꿨을 때 종료라는 텍스트가 있다면 소통이 종료 되도록 설계하였다.

2-5. 코드 최적화 경량화

우리의 목표는 PC환경에서만이 아니라 임베디드 보드위에서도 동작을 시키는게 목표이기 때문에 코드 자체를 경량화 최적화를 진행하여 보다 쫌더 수월한 동작이 이루어지게 진행하였다.

○ 최적화 부분
- 흑백화면 처리 및 출력 해상도를 낮춰 자원을 효과적으로 활용하여 성능을 향상, 소비되는 에너지나 시간을 최소화하도록 하였다.

○ 경량화 부분
- Thread라이브러리를 이용해 졸음감지 부분과 ChatGPT 소통이 병렬로 동작하도록 하여 최소한의 비용으로 최대한의 가치를 창출할 수 있게 하였다.

○ PC

PC환경에서는 성능이 좋아 영상이 끝어지지 않고 잘 작동하였고 최적화 경량화 진행을 한 후에 FPS 성능향상이 기존의 15~17에서 66~70까지 성능향상을 확인하였다.

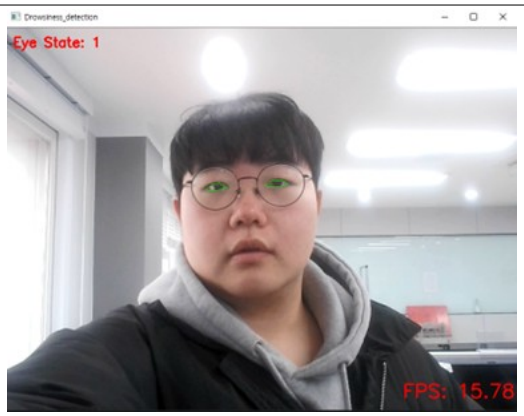


그림 17 PC환경 최적화, 경량화 전

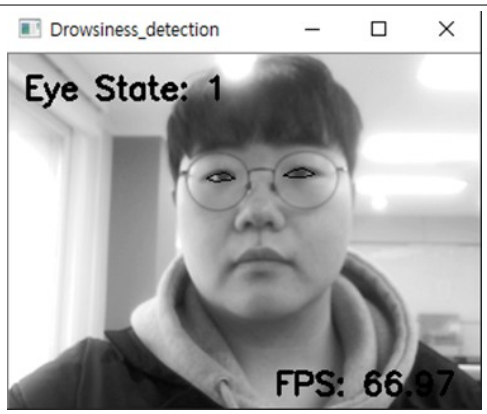


그림 18 PC환경 최적화, 경량화 후

○ **Jetson NANO**

Jeston NANO환경에서는 코드 최적화 경량화 전에는 영상이 4~6 FPS로 매끄럽지 못했지만 적용 후에는 17 FPS까지 올라고 PC환경만큼 매끄럽지는 못하지만 동작이 가능 할 수준으로 성능 향상된 걸 확인했다.

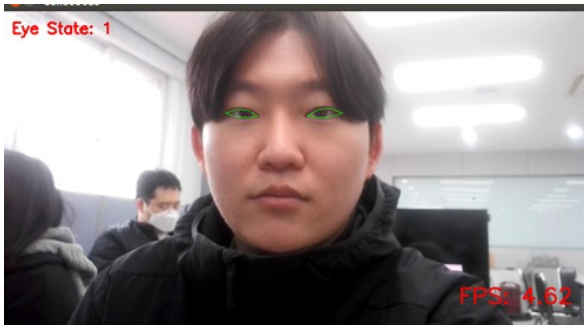


그림 19 Jetson NANO환경 최적화, 경량화 전

그림 20 Jetson NANO환경 최적화, 경량화 후

○ **Raspberry pi4**

Raspberry PI4환경에서는 최적화 경량화 전에는 7~8정도의 FPS를 성능을 보였는데 적용 후 25~26까지의 성능 향상을 보였다. 코드를 동작을 하고 수행 가능할 정도로 성능 향상이 되었고 임베디드 보드 환겨에서 동작가능을 확인하였다.

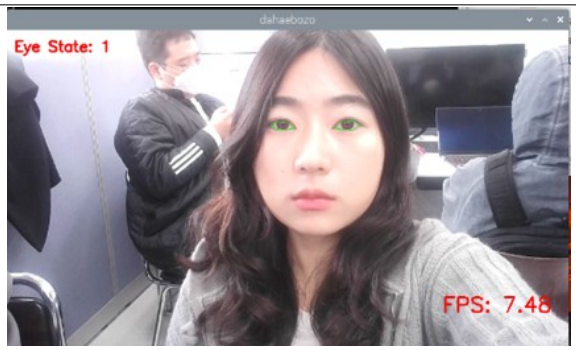


그림 21 Raspberry pi4환경 최적화, 경량화 전

그림 22 Raspberry pi4환경 최적화, 경량화 후

3) 결 론

사망사고로 이어질 확률이 큰 졸음운전을 하는 원인을 찾아보니 혼자 있을 때 발생을 하는데 그 이유

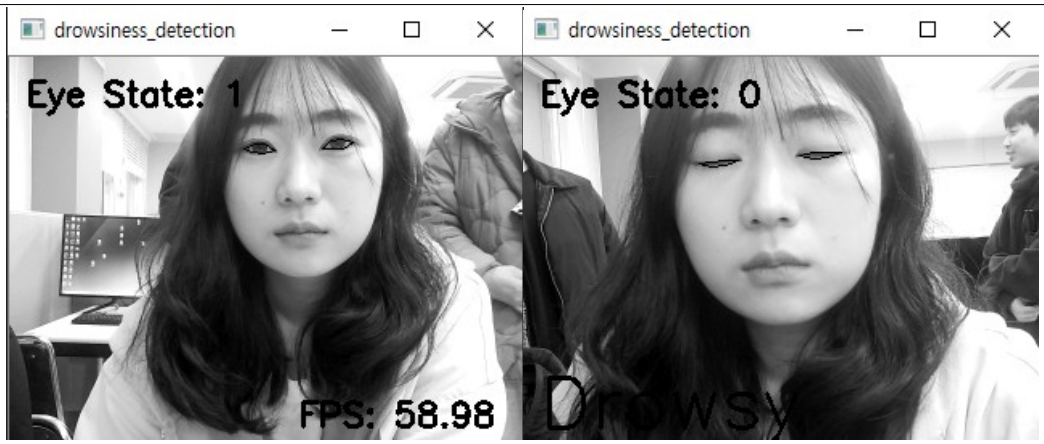
는 운전자의 주의를 환기시켜 줄 사람이 없어서 발생한다고 한다. 또 장시간 운전과 같은 장거리 운전, Co2농도의 증가에서 오는 집중력 저하도 요인이라고 한다. 찾아본 논문에 의하면 대화는 뇌를 활성화 즉 집중력을 향상 시킨다고 한다. 그래서 영상처리를 이용해 졸음감지 ChatGPT를 이용한 소통을 통해 운전자가 다시 졸음운전을 하지 않도록 감지와 도움을 주는 시스템을 개발하였다. 매주 팀 회의를 통해 우선순위를 결정하고 인원을 분배 했으면 팀 개발을 위한 개발 장소에 개발 환경을 구축하고 slack을 구축하여 보다 효율적으로 개발 진행하였다. 개발을 눈 감은 유무 판별, 졸음 감지, ChatGPT를 이용한 소통, TTS와 STT적용, 코드 경량화 최적화로 나누어 각 동작 확인을 한 후에 코드를 합쳐 오류가 발생하였을 때 쉽게 오류를 찾고 수정가능하게 진행하였다.

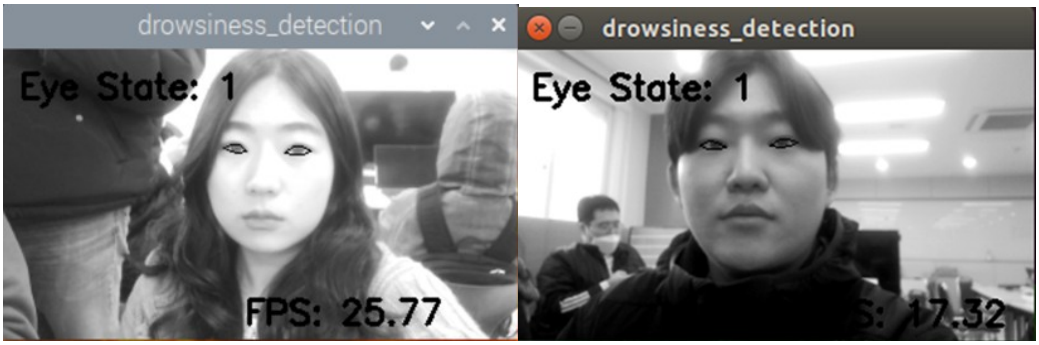
OpenCV, Dlib등 여러 라이브러리를 이용해 영상 처리를 통한 졸음 감지를 구현하였고 ChatGPT API키를 발급받아 TTS와 STT를 이용해 졸음운전을 하는 운전자와의 소통까지 구현을 하였다. 그 후 PC, Jetson NANO, Raspberry pi4에 동작 시켜보고 코드를 최적화 경량화를 통해 수월한 동작이 가능하게 성능을 향상시키고 동작 환경마다의 성능 차이를 확인하였다. 이러한 과정으로 제한된 자동차 공간에서도 임베디드 보드를 이용해 졸음운전을 하는 운전자를 영상처리를 이용해 졸음감지를 하고 ChatGPT를 이용해 운전자와의 소통을 통해 운전자가 다시 졸음운전을 하지 않게 감지와 도움을 주는 시스템 동작이 가능하다라는 걸 확인하였고 운전자에게 시스템을 제공해 줌으로써 졸음운전에서 벗어나 보다 안전한 운전을 기대할 수 있다.

6. 활용계획

○ 운전자의 운전 집중도를 향상 시켜 졸음 운전이 반복되지 않도록 도움을 준다.
○ 다양한 기기와 연동하여 졸음 감지시 소통외에도 다양한 졸음 방지 시스템을 제공할 수 있다.
○ TTS/STT를 이용해 자동화된 시스템을 제공할 수 있다.
○ Chat GPT를 이용한 소통은 운전자와의 소통뿐만 아니라 다양한 기기에 활용할 수 있다.

7. 실적물 사진






```
You said: 간단한 수학 문제 내 줘
간단한 수학 문제 내 줘
7 + 3
generating audio
speaking
speaking complete
PERCLOS: 0.1038961038961039
PERCLOS: 0.18142548596112312
result2:
{   'alternative': [{'confidence': 0.87048674, 'transcript': '정답은 10'}],
    'final': True}
You said: 정답은 10
정답은 10
맞았습니다!
```