

# 종합설계 프로젝트 수행 보고서

프로젝트명	Drowsiness Detection System
팀번호	S1-1
문서제목	수행계획서( ) 2차발표 중간보고서( ) 3차발표 중간보고서( ) 최종결과보고서( O )

2020.12.01.

팀원 : 정동원  
윤희영  
정지훈

지도교수 : 최진구 (인)

## 문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.08.	정 지 훈	1.0	수행계획서	최초작성
2020.05.02.	윤 희 영	1.1	본론4 추가	
2020.07.30	정 동 원	1.2	본론7 추가	
2020.11.28	정 지 훈	1.3	결론 추가	

## 문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6) II. 본론 (1~3) 참고자료	I. 서론 (1~6) II. 본론 (1~4) 참고자료	I. 서론 (1~6) II. 본론 (1~5) 참고자료	I. 서론 (1~6) II. 본론 (1~7) 참고자료	I II III

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계” 교과목에서  
프로젝트

“Drowsiness Detection System” 을 수행하는  
(S1-1, 정동원, 윤희영, 정지훈) 들이 작성한 것으로 사용하기 위해서는  
팀원들의 허락이 필요합니다.

# 목 차

## I. 서론

1. 작품선정 배경 및 필요성 .....
2. 기존 연구/기술동향 분석 .....
3. 개발 목표 .....
4. 팀 역할 분담 .....
5. 개발 일정 .....
6. 개발 환경 .....

## II. 본론

1. 개발 내용 .....
2. 문제 및 해결방안 .....
3. 시험시나리오 .....
4. 상세 설계 .....
5. Prototype 구현 .....
6. 시험/ 테스트 결과 .....
7. Coding .....

## III. 결론

1. 테스트 결과 및 개선 사항
2. 연구 결과 .....

참고자료 .....

# 서론

## 1. 작품선정 배경 및 필요성

필요성	내 용
수요배경	<ul style="list-style-type: none"> <li>• <b>졸음운전</b>의 경우 다른 교통사고에 비해 <b>치사율이 2배</b>이며, <b>매일 7건의</b> 졸음운전사고가 발생해 <b>14명의 사상자</b>가 발생함.</li> <li>• 졸음운전은 본인의 생명뿐만 아니라 무고한 타인의 생명을 치명적으로 위협한다.</li> <li>• 항공 등 중요 <b>관계 업무 수행</b> 시, <b>졸음</b>은 자칫하면 <b>큰 사고</b>로 이어줄 가능성이 매우 높음</li> <li>• 그러나 졸음 방지 기능을 가진 휴대용 장비 보급이 원활하지 않음.</li> </ul>
필요성 1	<ul style="list-style-type: none"> <li>• 실시간으로 <b>사용자의 얼굴</b>을 인식하고 졸음 여부를 판단함.</li> <li>• 얼굴인식과 더불어 <b>맥박을 측정</b>해 졸음 판단의 정확성을 향상시킴.</li> <li>• 졸음 판단 시 연결된 시스템을 통해 <b>경고음을 발생</b>시켜, <b>졸음 상태에서 벗어날</b> 수 있도록 도움을 줌. 이에 따라 <b>사고 예방에 큰 역할</b>을 할 것으로 기대됨.</li> </ul>
필요성 3	<ul style="list-style-type: none"> <li>• 현재 시중에 유통되는 졸음 방지 장비는 고가이며, 졸음운전에 국한된 것들이 대다수임</li> <li>• <b>저비용의</b> 간단한 장치만으로 <b>졸음 방지</b>를 가능케 할 <b>필요</b>가 있음.</li> <li>• 졸음운전의 상황뿐만 아니라, <b>졸음 방지가 필요한 다양한 분야에서 활용</b> 가능함</li> <li>• 자동차 산업의 확장에 따라 안전장치 역시 준비되어 있어야 함.</li> </ul>

## 2. 기존 연구/기술동향 분석

개발내용	내 용
국내 관련연구 소개 , 장점, 단점	<ul style="list-style-type: none"> <li>■ <b>심박변이도 측정을 통한 운전자 졸음 및 각성 상태 분석</b> <ul style="list-style-type: none"> <li>• (소개) 뇌전도, 심전도를 이용한 심박 변이도 분석</li> <li>• (장점) 심박도 뿐만 아니라 뇌전도와 심전도까지 활용하여 더욱 정확한 졸음 상태 파악이 가능하다.</li> <li>• (단점) 심전도 측정을 위해 매번 양 손목 및 왼 발목에 1회용 패치 전극을 부착해야 하는 3 lead wire 법을 이용했다.</li> </ul> </li> <li>■ <b>졸음 방지 시스템을 위한 눈 개폐 상태 판단 방법</b> <ul style="list-style-type: none"> <li>• (소개) 눈, 특히 동공 영상을 추출하고 이를 통해 운전자의 졸음을 판단하는 방법제안</li> <li>• (장점) Kinect를 사용한 실험 결과는 컬러 영상의 경우 100%였고 적외선 영상은 90%로 높은 정확도를 나타냄</li> </ul> </li> <li>■ <b>패치형 압력센서를 이용한 휴대용 실시간 무선 맥박 측정 시스템 개발</b> <ul style="list-style-type: none"> <li>• (소개) 정확한 혈압과 맥박을 측정하기 위해 L -C 공진을 이용한 측정시스템 개발</li> <li>• (장점) 반도체 공정을 이용한 압력센서를 제작하여 정확한 맥박 측정을 가능하게 했다.</li> <li>• (단점) 제작된 시제품의 휴대성이 불편하다.</li> </ul> </li> </ul>
국외 관련연구 특징, 장점, 단점	<ul style="list-style-type: none"> <li>■ <b>Heart Rate During Sleep: Implications for Monitoring Training Status, Miriam R . Waldeck</b> <ul style="list-style-type: none"> <li>• (소개) 수면 시와 활동 시 맥박을 모니터링 함</li> <li>• (장점) 맥박 측정에 대한 정확하고 많은 양의 정보가 담겨있음</li> </ul> </li> </ul>
공통적인 문제점	<ul style="list-style-type: none"> <li>■ <b>휴대성과 편리성의 부족</b> <ul style="list-style-type: none"> <li>• 기존의 졸음 방지시스템은 심전도 측정 기구를 부착해야 하는 등 졸음 방지를 위해 필요한 것들이 많다. 특히 운전의 경우 일상생활에서 자주 사용해야 하므로 그 무엇보다 편리성이 중요하다.</li> </ul> </li> </ul>

### 3. 개발목표

#### 최종목표

- 라즈베리파이 및 카메라 모듈을 사용, 사용자의 **상태를 파악**하여 졸음 및 부주의에 의한 **사고를 예방**하는 모듈을 개발

단계	내 용
1단계	<ul style="list-style-type: none"> <li>■ 운전자의 상태를 파악하기 위한 구현 모듈 <b>조사</b> <ul style="list-style-type: none"> <li>• 라즈베리파이를 활용한 심박도 체크 기능 조사</li> <li>• Adaboost 알고리즘 조사</li> <li>• Haar-like feature 기법 조사</li> <li>• 실시간 얼굴영역 검출을 위한 Open CV 라이브러리조사</li> </ul> </li> </ul>
2단계	<ul style="list-style-type: none"> <li>■ <b>실시간 얼굴 인식 기능 구현을 위한 아키텍처 설계</b> <ul style="list-style-type: none"> <li>• USB형 카메라와 라즈베리파이를 연동 시켜 실시간 얼굴 인식 프로세스 수립</li> <li>• 실시간 파악된 얼굴인식에 Haar-like feature와 Adaboost알고리즘을 적용</li> <li>• 졸음 상태라 판단될 경우, 알림 기능 설계</li> </ul> </li> </ul>
3단계	<ul style="list-style-type: none"> <li>■ 구현한 시뮬레이터에 대한 <b>테스팅 수행</b> <ul style="list-style-type: none"> <li>• 각 상황에 대한 <b>테스트 시나리오</b>에 대한 작성</li> <li>• 테스트 시나리오에 대한 <b>단위 테스트</b> 진행</li> </ul> </li> </ul>

### 4. 팀 역할 분담

	정동원	윤희영	정지훈
자료수집	머신 러닝 알고리즘	카메라 인터페이스, 졸음 알림 모듈	안면데이터 영역 검출 알고리즘
설계	머신 러닝을 이용한 졸음판단	카메라를 이용한 안면데이터 추출	안면데이터 -> 필요한 영역 검출
구현	데이터를 통하여 졸음판단 머신 러닝 구현	카메라를 이용한 안면데이터 수집, 졸음 알림 구현	OpenCV를 이용하여 머신 러닝 가능한 데이터로 변환
테스트	카메라 안면인식 테스트 졸음 판단 머신 러닝 테스트 시스템 통합 테스트		

## 5. 개발 일정

### ■ 추진 계획

추진 계획	12월	1월	2월	3월
과제기획	●			
요구사항 도출		●	●	
기본설계 및 상세설계		●	●	●

### ■ 개발 일정

활동 내용	12월				1월				2월				3월			
	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주
[과제기획]																
팀 구성	●															
업무분담	●	●														
아이디어 회의 및 정리		●	●	●												
개발 아이디어 선정			●	●												
과제명 선정				●												
[요구사항 도출]																
국내 동향 조사 및 분석				●	●	●										
해외 동향 조사 및 분석				●	●	●										
특징, 장점, 단점 분석					●	●	●									
문제점 정리						●	●	●								

문제점 개선을 위한 제안 사항 도출							●	●									
[기본설계 및 상세설계]																	
구현(개발)해야 할 내용 분석							●	●	●	●							
시나리오 설계								●	●	●	●						
시스템 구조 설계									●	●	●	●					
시스템 평가 방법 설계											●	●					
최종기획 보고서 완료												●	●				

## 5. 개발 환경

구 성	내 용
개발 도구	■ Python / QT-Designer
영상 처리 모듈	■ Open CV 라이브러리 이용 • Haarcascade
인공지능	■ Keras
Collaborator	■ Github • <a href="https://github.com/drowsiness-detection-system">https://github.com/drowsiness-detection-system</a>

## 본 론

### 1. 개발 내용

구 성	내 용
SW	<ul style="list-style-type: none"><li>■ 안면 데이터<ul style="list-style-type: none"><li>• 카메라를 통해 입력받은 영상 처리</li></ul></li><li>■ 실시간 얼굴 영역 검출기능 구현<ul style="list-style-type: none"><li>• Open CV 라이브러리 사용</li><li>• CascadeClassifier</li></ul></li><li>■ 정확도 향상을 위한 인공지능 학습법 사용<ul style="list-style-type: none"><li>• Keras 사용</li></ul></li></ul>
HW	<ul style="list-style-type: none"><li>■ 라즈베리파이4</li><li>■ 카메라 인터페이스</li><li>■ 알림 모듈</li></ul>

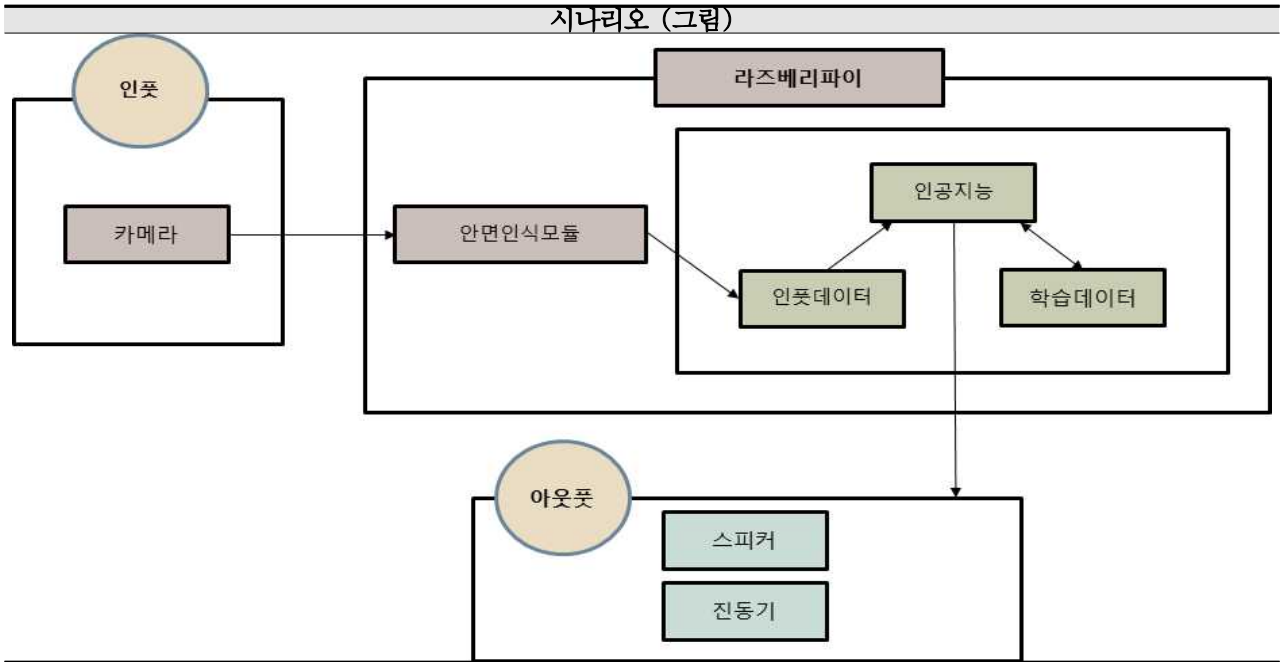
### 2. 문제 및 해결방안

순번	제안사항
1	<ul style="list-style-type: none"><li>■ <b>졸음 상태를 판단하는 시스템</b>을 제안<ul style="list-style-type: none"><li>• 보다 정확한 판단을 위하여, <b>안면인식</b> 수행</li><li>• 체크 된 얼굴인식 정보 활용</li></ul></li></ul>
2	<ul style="list-style-type: none"><li>■ 졸음 판단 시, <b>즉각적인 반응</b>을 통해 졸음을 방지 해 줄 수 있는 <b>졸음방지 시스템</b>을 제안<ul style="list-style-type: none"><li>• 스피커를 통한 <b>소리 알림</b> 제공</li></ul></li></ul>
3	<ul style="list-style-type: none"><li>■ <b>휴대성</b>이 우수한 졸음 방지 모듈을 개발, <b>다양한 분야에 적용</b> 가능한 시스템을 제안<ul style="list-style-type: none"><li>• 여러 분야에서 다양한 사용이 가능 하도록 모듈의 휴대성과 이식성에 중점을 둠</li></ul></li></ul>

### 3. 시험 시나리오

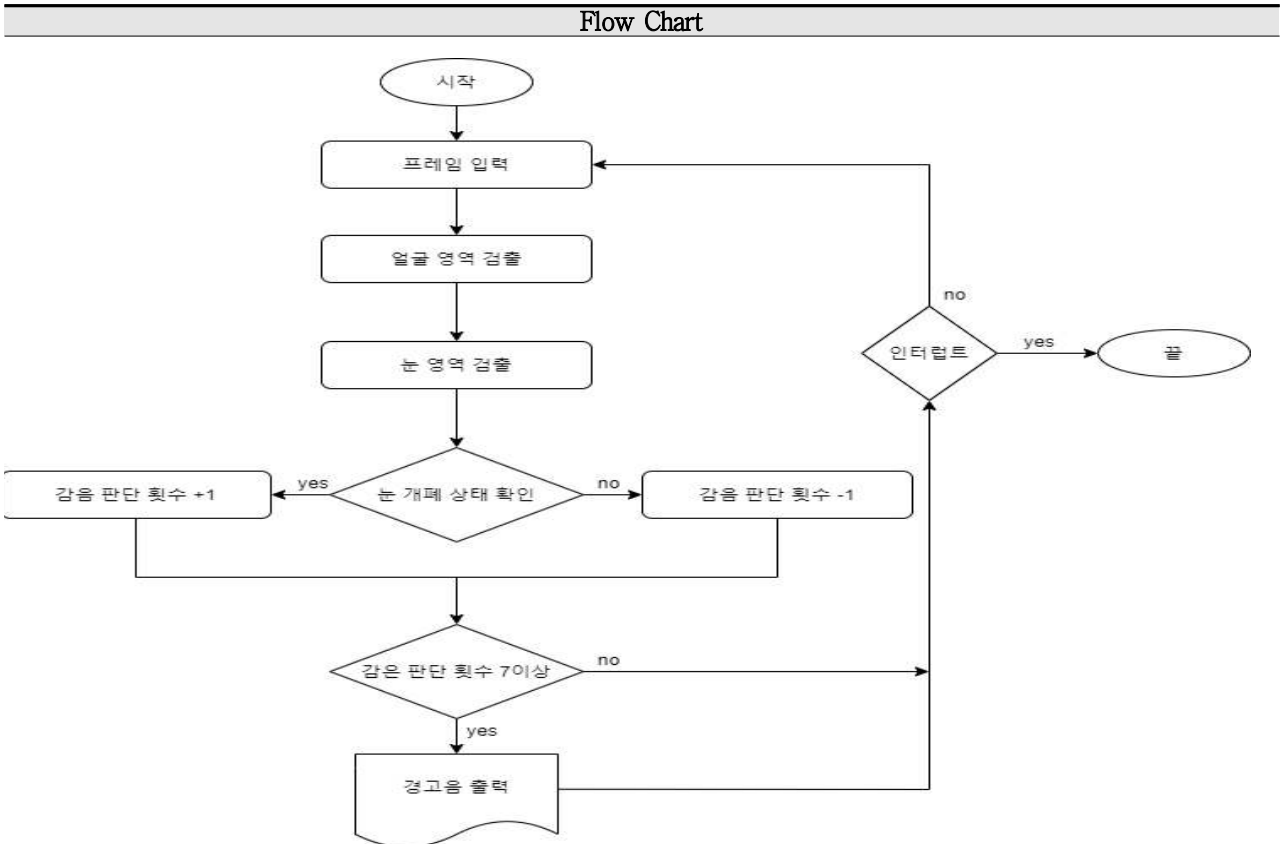
시나리오명	내 용
환경 감지	<ul style="list-style-type: none"><li>• 카메라를 통해 얼굴 정보가 정상적으로 입력되는지 확인함</li></ul>
데이터 저장	<ul style="list-style-type: none"><li>• 안면 인식 후 가공된 데이터를 저장함.</li><li>• 데이터를 머신 러닝을 통해 졸음 상태 정보 추적</li></ul>
사용자 상태 변화	<ul style="list-style-type: none"><li>• 사용자가 졸음 상태로 변화될 경우, 이를 감지하여 알림 센서에 동작 정보 전달.</li></ul>
알림센서 구동	<ul style="list-style-type: none"><li>• 동작 정보를 받은 알림 센서는 즉시, 알림 기능 수행</li></ul>





#### 4. 상세 설계

##### ■ 플로우 차트



## 서비스용 UI 개발


- 이용 방법이 비교적 쉬우며, 직관적인 인터페이스를 가진 GUI 편집기인 Qt Designer를 사용하여ui 설계
- 로고는 “Janie”를 사용.

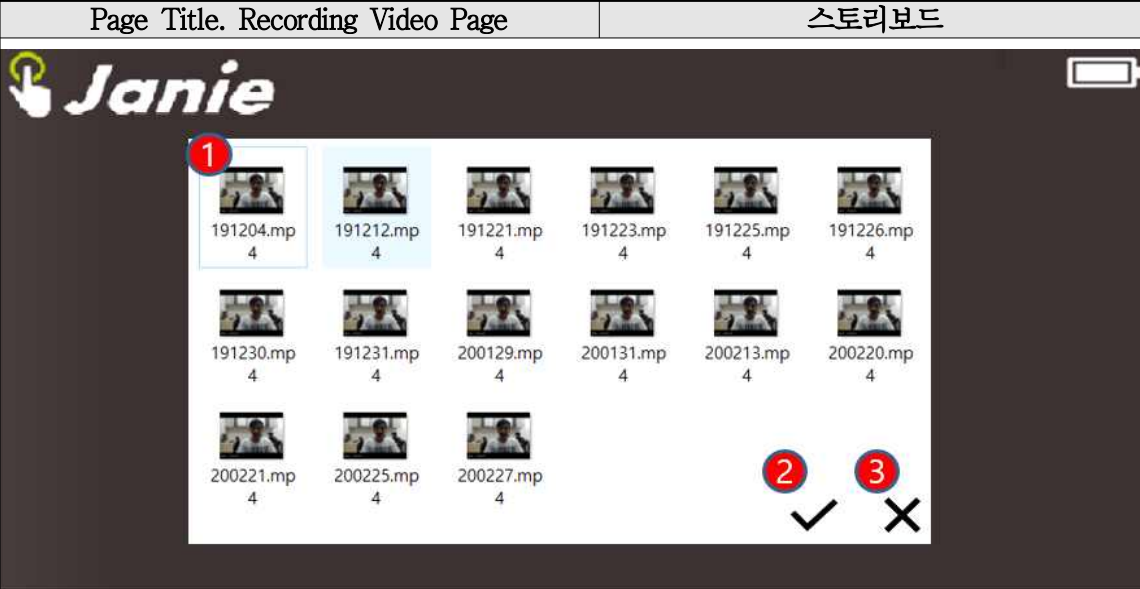


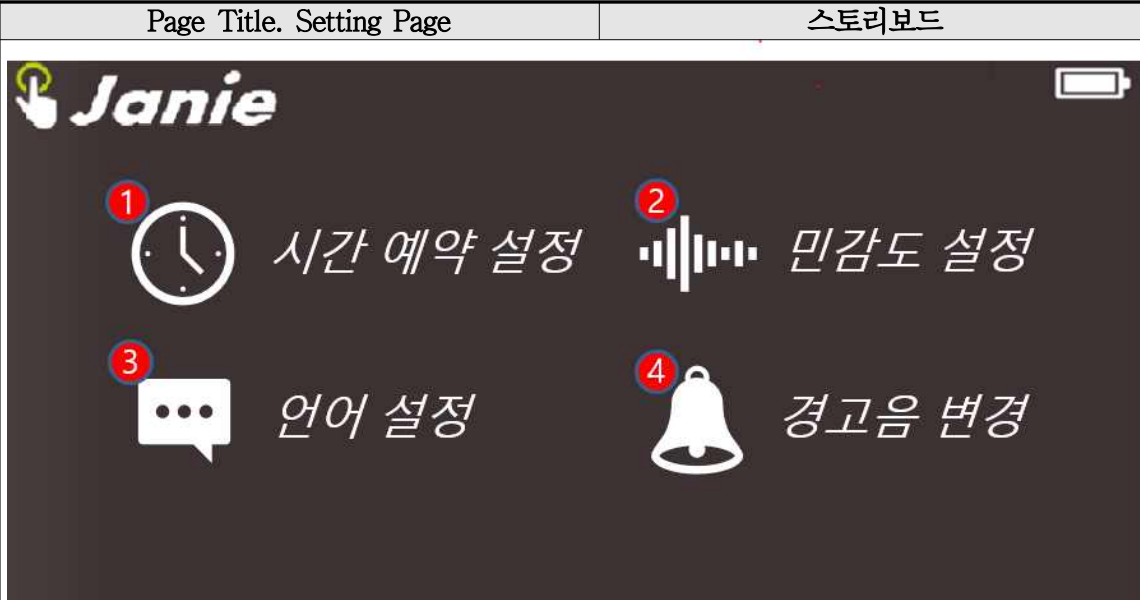
### ■ 본 프로그램 스토리 보드

Page Title. Main Page		스토리보드
페이지설명	초기 실행 시 보이는 대표 화면	
1	최근 로그 및 기록 확인 페이지로 이동	
2	개발자 정보 페이지로 이동	
3	졸음 감지 모드 동작 화면으로 전환	
4	기록된 영상 확인	
5	각종 설정 셋팅 화면으로 전환	
6	프로그램 종료 버튼	
Page Title. Reports Page		스토리보드
페이지설명	졸음상태라고 판단될 때마다 기록함	
1	메인페이지로 이동	

Page Title. Support Page		스토리보드
		
페이지설명	개발자 정보 및 연락처	
1	메인페이지로 이동	

Page Title. Running Page		스토리보드
		
페이지설명	졸음 감지 시스템 동작 화면	
1	메인 메뉴 화면으로 이동	
2	현재 비 졸음 상태 표시	
3	배터리 잔량 표시	
4	실행 버튼	
5	일시 정지 버튼	
6	경고음 음량 설정	

Page Title. Recording Video Page	스토리보드
	
페이지설명	녹화된 영상 목록 및 재생 화면
1	해당 영상 플레이
2	영상들 선택 버튼
3	선택된 영상 삭제 기능 버튼

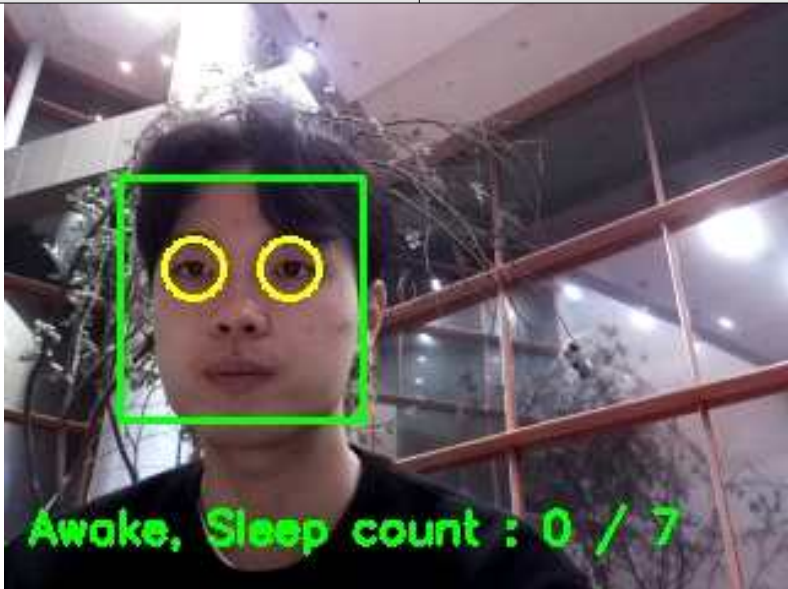
Page Title. Setting Page	스토리보드
	
페이지설명	각종 설정 변경 셋팅 화면
1	프로그램 종료
2	메인메뉴로 이동
3	프로그램 종료
4	메인메뉴로 이동

Page Title. Exit Page		스토리보드
		
페이지설명	프로그램 종료 여부 재확인 화면	
1	프로그램 종료	
2	메인메뉴로 이동	

구성	S W 상 세 설 계
1	<b>■ 서비스용 UI 모듈</b> <ul style="list-style-type: none"> <li>파이썬의 QT Designer 프로그램을 이용하여 개발</li> <li>직관적이며 총 6개의 버튼으로 메뉴 구성</li> <li>동작 / 통계 기록 / 설정 / 영상 확인 / 종료 등으로 구성</li> </ul>
2	<b>■ 프레임 입력 모듈</b> <ul style="list-style-type: none"> <li>라지베리파이카메라와 본 프로그램 간의 연동</li> <li>높이 480px / 너비 640px / 초당 16프레임 캡처</li> </ul>
3	<b>■ 얼굴영역 및 눈 영역 검출 모듈</b> <ul style="list-style-type: none"> <li>openCV의 CascadeClassifier()에서 'Haarcascade_frontalface_alt.xml'과 'Haarcascade_eye_tree_eyeglasses.xml' 두 파일을 이용하여 얼굴과 눈 영역을 각각 검출 한다.</li> </ul>
4	<b>■ 눈 개폐 상태 확인 모듈</b> <ul style="list-style-type: none"> <li>인공지능 알고리즘인 Neural Network (Keras) 모듈을 활용</li> <li>"Data Flair"사이트에서 배포하는 Keras Deep Learning 모델을 로딩</li> <li>로딩 된 모델을 이용하여 눈 개폐 여부 판단</li> </ul>
5	<b>■ 졸음 상태 판단 및 경고음 출력 모듈</b> <ul style="list-style-type: none"> <li>일정 기준 초과 시 경고음 발생</li> <li>"Pygame 라이브러리"의 Mixer를 사용하여 경고음출력</li> </ul>

구성	H W 상 세 설 계
1	■ 라즈베리파이4
2	■ 라즈베리 카메라 모듈
3	■ 라즈베리파이 7인치 정전식 터치 LCD
4	■ 스피커

## 5. Prototype

Prototype example	진행 과정 1
	
1	사용자의 얼굴 영역과 눈 영역 검출 후, 눈의 개폐 여부 판단
2	프로그램이 사용자가 비 수면이라 판단 시, Sleep count가 프레임 당 1 씩 감소 ( 단, Sleep count는 최소 0 )

Prototype example		진행 과정 2
		
1	사용자의 얼굴 영역과 눈 영역 검출 후, 눈의 개폐 여부 판단	
2	프로그램이 사용자가 수면이라 판단 시, Sleep count가 프레임 당 1 씩 증가	
3	하단의 글씨 역시 붉은색으로 변화	

Prototype example		진행 과정 3
		
1	사용자의 얼굴 영역과 눈 영역 검출 후, 눈의 개폐 여부 판단	
2	Sleep count가 7 이상이 될 경우, 화면이 흑백으로 변화	
3	동시에 알람모듈을 통해 경고음 출력	

## 6. 시험/ 테스트 결과

순 번	평가방법
01	<ul style="list-style-type: none"> <li>• 한쪽 눈만으로도 기능이 정상 작동하는지 확인함.</li> </ul>
02	<ul style="list-style-type: none"> <li>• 인식 대상이 프레임의 어느 곳에 있든 인식함을 확인함.</li> </ul>
03	<ul style="list-style-type: none"> <li>• 안경 착용 시, 정확히 작동 되는지 확인함.</li> </ul>
04	<ul style="list-style-type: none"> <li>• 야간 환경에서도 기능이 정상 작동 되는지 확인함.</li> </ul>

## 7. Coding

### Coding (핵심 기능)

```

from picamera.array import PiRGBArray
from picamera import PiCamera
import cv2
import numpy as np
import tensorflow as tf
import pygame
from tf.lite_runtime.interpreter import Interpreter
pygame.mixer.init()
pygame.mixer.music.load('alarm.wav') #졸음판단시, alarm 파일 실행
face_cascade_name = './haarcascades/haarcascade_frontalface_alt.xml'
eyes_cascade_name = './haarcascades/haarcascade_eye_tree_eyeglasses.xml'
face_cascade = cv2.CascadeClassifier()
eyes_cascade = cv2.CascadeClassifier()

```



---

```

        # 안면과 눈 영역 검출을 위한 파일 로드
    if not face_cascade.load(cv2.samples.findFile(face_cascade_name)):
        print('--(!)Error loading face cascade')
        exit(0)
    if not eyes_cascade.load(cv2.samples.findFile(eyes_cascade_name)):
        print('--(!)Error loading eyes cascade')
        exit(0)
    #tflite모델 로드
    interpreter = Interpreter('drowsinessDetection.tflite')
    interpreter.allocate_tensors()
    SZ = 24
    status = 'Awake'
    number_closed = 0
    closed_limit = 7 # number_closed 7 초과 시, 알람
    max_count = 10 # 최대10까지만 증가하도록 설정
    show_frame = None
    sign = None
    color = None
    frame_width = 320
    frame_height = 240
    frame_resolution = [frame_width, frame_height]
    frame_rate = 32
    def set_input_tensor(interpreter, image):
    tensor_index = interpreter.get_input_details()[0]['index']
    input_tensor = interpreter.tensor(tensor_index())[0]
    input_tensor[:, :] = image
    #tflite모델을 이용해 눈 개폐 여부 판별하는 함수
    def classify_image(interpreter, image, top_k=1):
    """Returns a sorted array of classification results."""
    set_input_tensor(interpreter, image)
    interpreter.invoke()
    output_details = interpreter.get_output_details()[0]
    output = np.squeeze(interpreter.get_tensor(output_details['index']))
    if output_details['dtype'] == np.uint8:
    scale, zero_point = output_details['quantization']
    output = scale * (output - zero_point)
    ordered = np.argsort(-output, top_k)
    return [(i, output[i]) for i in ordered[:top_k]]
    # 라즈베리파이 카메라 초기화 및 설정
    camera = PiCamera()
    camera.rotation = 90
    camera.hflip = True
    camera.resolution = frame_resolution
    camera.framerate = frame_rate
    rawCapture = PiRGBArray(camera, size=(frame_resolution))
    # 카메라로 입력받은 프레임마다 수행
    for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array
        show_frame = image

```

---

---

```

        # 졸음판단시, 흑백 화면 송출
        height, width = image.shape[:2]
        frame_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        frame_gray = cv2.equalizeHist(frame_gray)

        faces = face_cascade.detectMultiScale(frame_gray) # 얼굴영역 검출
        for (x, y, w, h) in faces:
            show_frame = cv2.rectangle(show_frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            faceROI = frame_gray[y:y + h, x:x + w]

        eyes = eyes_cascade.detectMultiScale(faceROI) #검출된 얼굴영역 중, 눈 영역 검출
        results = []
        for (x2, y2, w2, h2) in eyes:
            eye_center = (x + x2 + w2 // 2, y + y2 + h2 // 2)
            radius = int(round((w2 + h2) * 0.25))
            show_frame = cv2.circle(show_frame, eye_center, radius, (0, 255, 255), 2)
            eye = faceROI[y2:y2 + h2, x2:x2 + w2]
            eye = cv2.resize(eye, (SZ, SZ))
            eye = eye / 255
            eye = eye.reshape(SZ, SZ, -1)
            eye = np.expand_dims(eye, axis=0)
            prediction = classify_image(interpreter, eye)
            result, confidence = prediction[0]
            results.append(result)
            if (np.mean(results) == 1):
                color = (0, 255, 0)
                status = 'Awake'
            number_closed = number_closed - 1
            if (number_closed < 0):
                number_closed = 0
            else :
                color = (0, 0, 255)
                status = 'Sleep'
            number_closed = number_closed + 1
            if (number_closed > max_count):
                number_closed = max_count

        sign = status + ', Sleep count : ' + str(number_closed) + ' / ' + str(closed_limit)
        if (number_closed > closed_limit):
            show_frame = frame_gray # 흑백화면 송출

        # get_busy() 반환값이 False일 때까지, 계속 알람 실행
        if (pygame.mixer.music.get_busy() == False):
            pygame.mixer.music.play()

        # 프레임 출력
        cv2.putText(show_frame, sign, (10, height - 20), cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
        cv2.imshow("Drowsiness Detection2", show_frame)

```

---

---

```
# 다음 프레임을 위해 스트림 초기화
rawCapture.truncate(0)
```

```
# 종료
key = cv2.waitKey(1) & 0xFF
if key == ord("q"):
    break
```

---

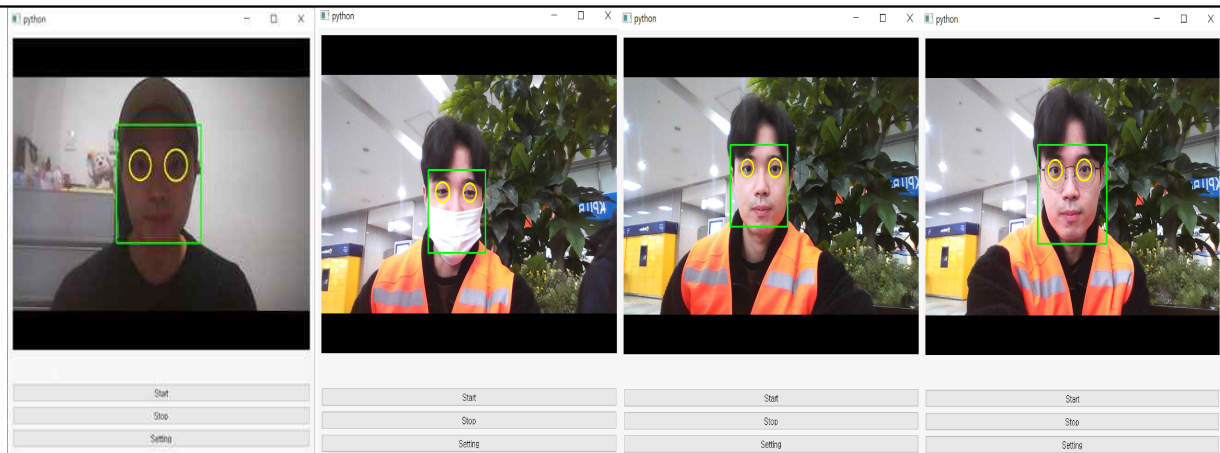
## 결론

### 1. 테스트 결과 및 개선 사항

#### 테스트 결과

본 시스템의 구현을 마치고 안경과 모자, 마스크를 착용한 상태에서의 테스트를 진행 했으며, 조도의 차이 역시 고려하여 테스트를 진행하였다.

테스트 결과 그림[1]과 같이 다양한 조건에서 안면영역을 정확히 검출한 후 졸음 판단 시에는 화면이 흑백처리 되며, 사용자가 졸음 상태에서 벗어날 수 있도록 경고음을 정상적으로 출력하였다. 마스크를 착용한 경우 얼굴영역 검출의 정확성이 떨어지는 문제점은 존재하였으나 졸음판단은 가능함을 확인하였다. 이로써 초기 구현 목표인 저비용의 높은 성능을 가진 졸음인식 시스템을 구현하였다.



<모자착용 테스트>

<마스크 테스트>

<야외조건 테스트>

<안경착용 테스트>

그림 1. 다양한 조건에서의 테스트 결과

---

## 개선 사항

추가적으로 진행한 테스트에서 얼굴영역 검출 과정에서 초기의 haarcascade\_frontalface\_default.xml 이 용하였을 때, 얼굴 측면을 인식하지 못하는 현상이 발견되었다. 이 문제점을 개선하기 위해 기존의 haarcascade\_frontalface\_default.xml이 아닌 dlib의 face detector를 사용하여 해결하였고 얼굴 측면을 정상적으로 인식함을 확인 할 수 있었다.

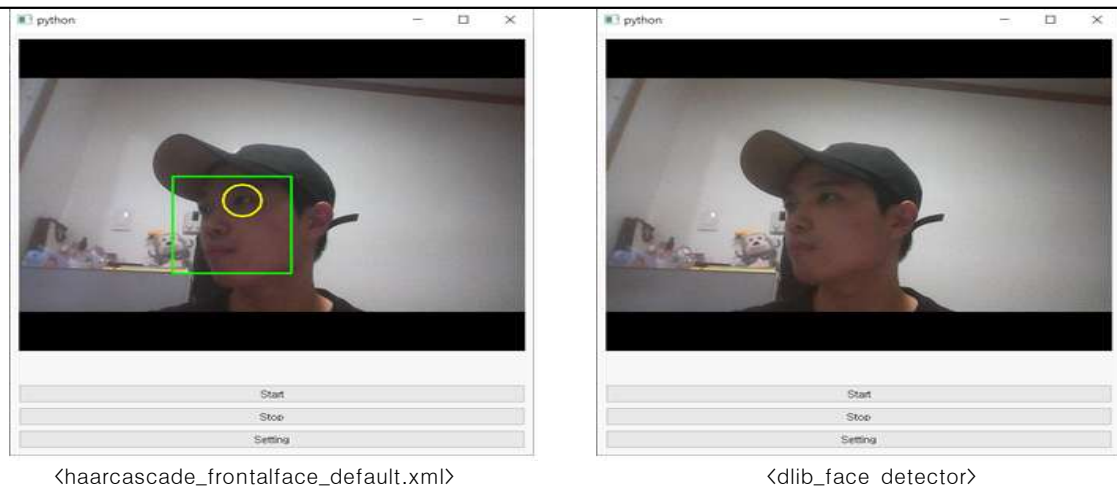


그림 2. haarcascade\_frontalface\_default.xml, dlib 안면 인식 차이

## 2. 연구 결과

### 결론 및 향후 연구 방향

기존에 개발된 고비용의 기술들을 대체할 수 있는 저비용의 높은 성능을 가진 줄음 감지 시스템을 구현하였다. 성능을 높이기 위해 CNN 모델을 사용하였으며 이를 통해 저가의 제품에서 인식률이 떨어지는 문제점을 해결하였다. 해당 모델을 Tensorflow Lite Converter를 통해 tflite모델로 경량화 시켜 라즈베리 파이와 같은 초소형/초저가의 컴퓨터에서 실행이 가능하며 운전 및 항공 관제 등에서 특별한 장치나 고가의 기술 없이도 간단히 설치하여 사용할 수 있을 것으로 예상된다.

향후 실시될 연구에서는 눈 크기의 편차를 고려하고 이를 극복하기 위한 방법과 야간주행과 같은 다양한 환경을 고려해 연구를 진행할 계획이다. 또한 측면 인식률을 개선하기 위해 dlib 라이브러리의 face detector를 사용하였음에도 불구하고 측면의 인식률이 떨어지는 모습을 보여 cvlib을 적용하거나 DNN face detector를 적용해 개선하는 것을 목표로 연구를 진행 할 것이다.

---

## 참 고 자 료

### <참 고 문 헌>

- 1) 김기훈. “고속도로 교통사고 사망자 10명 중 6~7명은 졸음운전·주시태만” 「연합뉴스」, 2020.07.23
- 2) 김원기 “졸음운전 치사율 과속사고의 2배” 「뉴스프리존」, 2016.02.06.
- 3) “또 '공항 관제사 졸음 사고'...리노서 심야에 무허가 착륙” 「LA중앙일보」, 2011.04.14.
- 4) 이원영, 오주석 ‘졸음운전 방지를 위한 대책에 관한 연구’ 연구보고서 2014-0107-105

### <인용 데이터 및 출처>

- 1) yawn\_eye\_dataset\_new For drowsiness detection (<https://www.kaggle.com/serenaraju/yawn-eye-dataset-new>)
  - 2) ML-Sigmoid 대신 ReLU? 상황에 맞는 활성화 함수 사용하기  
(<https://medium.com/@kmkgabia/ml-sigmoid-%EB%8C%80%EC%8B%A0-relu-%EC%83%81%ED%99%A9%EC%97%90-%EB%A7%9E%EB%8A%94-%ED%99%9C%EC%84%B1%ED%99%94-%ED%95%A8%EC%88%98-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0-c65f620ad6fd>)
  - 3) Face detection using dlib HOG (<https://medium.com/mlcrunch/face-detection-using-dlib-hog-198414837945>)
-