

# 개발 완료 보고서

제출일 : 2025년 3월21일

프로젝트 소개			
팀명	임김록		팀원 김도엽, 임정규
활동일시	03 / 13 ~ 03 / 21		장소 공학1관 드론융합실
프로젝트 명	DD (Don't be Drowsy)		
주요 주제	웹캠 기반 자유 프로젝트		
개발목적	딥러닝 이미지 인식을 이용한 운전자의 졸음 운전 방지		
개발환경	Windows 11 / C# WPF .NET Core / Visual Studio 2022 / MySQL / Python 3.12		
구현기능	김도엽	1. 웹캠 연결 및 이미지 저장 2. 클라이언트 UI 구성 3. YOLO 모델을 이용한 졸음 판정 로직 작성	
	임정규	1. 이미지 전송 프로토콜 작성 2. 모델 선별, YOLO 모델 학습 및 적용 3. Database 연동 및 Log 저장	
메서드 설명	클라이언트 (C#)	<b>1. Window_Closed</b> <ul style="list-style-type: none"><li>클라이언트 창 종료시 웹캠도 종료</li></ul> <b>2. startButton_Click</b> <ul style="list-style-type: none"><li>비디오 장치 탐색 후, 첫 번째 비디오 장치에 연결</li></ul> <b>3. closeButton_Click</b> <ul style="list-style-type: none"><li>비디오 장치 실행 중이라면, 연결 종료</li><li>서버와 연결되어있는 상태라면, 서버와의 연결 종료</li></ul> <b>4. video_NewFrame</b> <ul style="list-style-type: none"><li>비디오 프레임 비트맵으로 변환</li><li>웹캠 영상 UI에 표시</li></ul> <b>5. BitmapToBitmapImage</b> <ul style="list-style-type: none"><li>비트맵으로 변환했던 이미지 토대로 캡처 이미지 저장</li></ul> <b>6. connectButton_Click</b> <ul style="list-style-type: none"><li>버튼을 누르면 서버와 연결</li><li>타이머 함수 작동, 0.5초마다 웹캠 영상 캡처해서 서버로 보냄</li></ul> <b>7. timer_Tick</b> <ul style="list-style-type: none"><li>캡처한 이미지 서버에 전송</li><li>서버로부터 메시지 수신 받아서 UI에 출력</li></ul> <b>8. captureImage</b> <ul style="list-style-type: none"><li>캡처한 이미지 지정해둔 경로에 저장하는 메소드</li></ul>	

메서드 설명	서버 (Python)	<h3>1. main</h3> <ul style="list-style-type: none"> <li>• 소켓 생성 및 C# 클라이언트와의 연결 대기</li> <li>• 연결 후 새로운 스레드를 통해 client_handler() 메서드 호출</li> <li>• 기존 스레드는 다시 accept로 넘어가서 새로운 연결 대기</li> </ul> <h3>2. client_handler</h3> <ul style="list-style-type: none"> <li>• 이미지를 클라이언트로부터 수신</li> <li>• model.predict() 실행 후 Model/predict_folder/predict_result/에 저장</li> <li>• 감지된 객체의 클래스 확인 (Awake, Drowsy)</li> <li>• 줄음 스택(drowsy_stack) 증가 조건 <ul style="list-style-type: none"> <li>- Drowsy 상태 감지되면 +1</li> <li>- 모두 Awake면 drowsy_stack = 0</li> </ul> </li> <li>• 판정 결과를 클라이언트로 전송 후 data_scrapper() 메서드를 통해 저장</li> </ul> <h3>3. data_scrapper</h3> <ul style="list-style-type: none"> <li>• MySQL에 연결 후 테이블이 존재하지 않을 경우 테이블 생성</li> <li>• 로그(접속 IP, 현재 시각, 판정 결과)를 Database에 저장</li> </ul>
개발후기	김도엽	<p>가상환경 세팅이 예상보다 까다로웠다. 단순히 해당하는 버전만 맞추면 될 줄 알았는데, 원하는 버전이 없거나 버전마다 설정 방법이 달라지기도 해서 일일이 찾아가며 해결해야 했다.</p> <p>또한, 인공지능 모델을 학습시키는 과정도 쉽지 않았지만, 팀원의 도움 덕분에 완성도 높은 모델을 구축할 수 있었고, 그로 인해 팀 프로젝트답게 제대로 진행이 된 것 같아 만족스러웠다.</p> <p>가상환경 세팅, 인공지능 모델 사용, 서버/클라이언트 통신 등 모든 과정이 쉽지는 않았지만, 결국 잘 해냈다는 점에서 큰 성취감을 느꼈고 이번 프로젝트를 통해 한 단계 더 성장한 기분이다.</p>
	임정규	<p>평소에 인공지능을 접하면서 인공지능의 성능에 놀라는 일이 많았으나, 한편으로는 인공지능이 생각만큼 대단하지 않단 얘기도 종종 들곤 했다. 이번에 몇몇 모델을 사용해 보면서 생각보다 낮은 성능에 역시 인공지능은 믿지 못하겠단 생각을 하기도 하고, YOLO의 우수한 성능에 놀라면서 세상의 발전에 감탄하기도 하는 며칠이었다.</p> <p>새삼 어떠한 인공지능을 다룰 수 있는지, 어떤 방식의 사용 방법을 익히는지에 따라 결과물의 차이가 천차만별이 될 것이란 것을 몸소 경험할 수 있었다. 또한 신기술일수록 자료가 모자랐기에 더욱 많은 자료를 찾고 공부를 지속해야 한다는 것을 자연스럽게 체감할 수 있었다. 나아가서 인공지능 학습을 위해서는 하드웨어 성능이 받쳐줘야 함도 배울 수 있었기에, 최신 기술에 필요한 것이 어떤 것들일지 익힐 수 있는 프로젝트였다. 상당히 흥미로운 한 주가 되었으며 파이썬과 C# 간의 네트워크 및 파일 전송을 완벽히 해내어 여러 프로젝트를 통해 네트워크에 관해서는 다양하게 고민하고 해결할 수 있었고, 어려웠던 네트워크에 대한 자신감이 생겼다.</p>

# 요구사항 명세서

유형	세부 분류	요구 분석 내용	비 고
클라이언트	줄음 운전 방지	C# WPF UI와 웹캠을 연동	
		연결된 웹캠으로부터 수신된 영상을 UI에 실시간으로 출력	
		서버로부터 수신된 판단 결과에 따라 경보 여부 결정	
	네트워크	웹캠 영상을 0.5초마다 캡처해서 서버로 송신	
서버	네트워크	클라이언트로부터 0.5초마다 이미지 수신	
	줄음 운전 여부 판단	클라이언트로부터 수신된 이미지를 인공지능 모델에 대입	
		인공지능 모델이 줄음 여부 판단	
		판단 결과를 클라이언트로 송신	
	DB	판단 결과 데이터를 MySQL DB에 저장	
인공지능	YOLO	데이터셋에 들어있는 이미지, CSV 파일 학습	

