캡스톤디자인 멘토링 보고서 (3차)

# 팀 기본 정보

|  |  |  |  |
| --- | --- | --- | --- |
| **분반** | **조원 명단** | **멘토** | **멘토링 일시** |
| 2 분반 | 현윤성, 안재현, 황재현 | 박소연 | 2025년 05월 08일 |
| **프로젝트 주제명** | TAKE A LOOK (웹캠 제스쳐 인식 및 자세 개선 어플리케이션) | | |

# 교수님 피드백 결과 및 반영 계획

|  |
| --- |
| 교수님 피드백 결과 |
| * **타겟층**  1. 연령대  * 유치원 (미취학 아동) 연령대는 어림.   + - * **랩탑 사용 중인 초중고 학생**  1. 환경  * 하드웨어 지원으로 인한 ‘모바일 vs 랩탑’ 환경 선정 고민   + - * **타겟층이 확고한 랩탑 환경** * **기능**   + - 저조도 환경의 필요성 (타겟 연령대의 큰 이슈 – 몰컴)     - 부모의 얼굴을 띄우는 피드백 (강제성 및 효과 극대화) |

|  |
| --- |
| 교수님 피드백 반영 계획 |
| **1. 타겟 환경 및 그에따른 연령층 수정**   * 타겟층이 확고한 랩탑 환경으로 확정 * 타겟 환경 (랩탑)에 맞는 연령대 수정   + - * **노트북을 사용하는 초중고 학생**   **2. 기능 우선순위 설정**   * 저조도 환경 판별 및 피드백 |

# 멘토링 결과 및 반영 계획

|  |
| --- |
| 멘토링 결과 |
| * **기능 구현**  1. **눈 깜빡임**  * **제한사항 -** 현실적으로 매우 높은 난이도 (실제 현업 종사자의 경험)  1. 눈이 작은 사람 2. 안경 쓴 사람 3. 저조도 환경 4. 잠깐 시야 돌리기  * **랜드마크 인식 모델 사용** * onnx 모델 (opensource model: openpose) * 해당 onnx 모델은 전신의 관절 포인트를 인식하는 구조 * 정확도가 낮아 상반신만 인식하도록 학습하는 방안   + - * **(답변) 2주만에 만들어서 데모 시연은 어려울 것** * mediapipe (Google) * Python으로 모든 MVP의 작동 여부 확인 완료 * Swift에서 사용하기 위한 C++ 마이그레이션 과정에서 오류 발생   + - * **(답변) tflite to onnx 변환이 가능하므로 일정 수준의 정확도 보장 가능** |

|  |
| --- |
| 멘토링 반영 계획 |
| * **정확도** * 눈 깜빡임 인식 정확도 향상 * 판단 알고리즘 개선, 추정값에 대한 신뢰도 도입 * 인간의 얼굴, 사람, 의류, 스크린샷이(가) 표시된 사진    자동 생성된 설명기능 구현 계획 * *하단 ‘진행상황’ 파트에서 서술 예정*   Google Meet 온라인 회의 중 |

# 진행상황

|  |
| --- |
| 진행상황 |
| * **사용자 피드백** * ChatGPT 활용, 사용자에게 제공할 기본 피드백 캐릭터 생성 * **계획 확장** * 현재 이슈  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **모델** | **Run on Python** | **C++ Migration** | **Speed** | **Accuracy** | **Run on Swift** | | onnx (tflite) | Yes | Yes | Fast | Bad | No | | mediapipe | Yes | No | Fast | Good | No |  * 계획 설정 및 확정  1. **Plan A**  * **Swift ⇔ Python 내장 소켓 통신 (구현 완료)**  1. ~~Plan B~~  * ~~Mediapipe C++에서 sdk화 ⇒ Swift~~  1. ~~Plan C~~  * ~~Swift pythonkit 에서 직접 파이썬 모델 실행 (라이브러리 import 매우 불안정)~~ |
| * **Swift – Python 내장 소켓 통신** * **Python에서 mediapipe 구동**   + - * 빠른 처리 속도, 안정성, 정확성 * **Swift에서 피드백 처리**   + - * 네이티브 UI, 전체화면 및 투명도 피드백 가능 * **Rough FlowChart**      * **개발 완료해야할 기능** * 안재현 (Swift) * 소켓 통신 최적화 * 프론트엔드 피드백 화면 구성 * 황재현 (Python) * 자세 판단 알고리즘 개발 * 현윤성 (Python) * 눈 깜빡임 추정 및 신뢰도 알고리즘 * 자세 유지 시간 측정 -> 데이터화 |