

1. 서론

1.1 개발 과제의 개요

가. 개발 과제 요약

어르신들의 건강 및 안전 케어에 대한 관심이 높아지고 있으며, 이를 위한 다양한 서비스가 검토되고 있다. 특히 어르신들의 거주 공간에서 생활 정보를 바탕으로 어르신들의 응급 안전 대응 및 건강 관리를 할 수 있다면 이는 매우 효과적인 서비스가 될 것이다. 본 개발의 목표는 디스플레이형 AI 단말기를 활용하여 65세 이상의 시니어를 대상으로 일상 생활 데이터를 모으고, 앞에서 말한 서비스 중 실시간 행동 분석을 통한 어르신들의 안전 케어 및 위험 상황 판단에 대한 스마트 서비스를 제공하는 시스템을 개발하는 것이다.

나. 개발 과제의 배경 및 효과

◇ 개발 과제의 배경

1. 시니어 케어 관련

2025년경 우리나라는 65세 이상 고령 인구가 총인구에서 차지하는 비율이 20%에 이르는 초고령 사회로 진입할 것으로 예상된다. 이에 맞춰 각 기업은 음성 인식, 객체 탐지, 자연어 처리같은 딥러닝 기술들을 활용한 '시니어 케어 서비스'를 선보이며 노년층 수요 잡기에 나서고 있다. 한글과컴퓨터는 '한컴 말랑 말랑 행복케어'를 통해 인지훈련 치매예방 가상현실(VR)과 상호교감이 가능한 AI 로봇 활용 프로그램을 개발 중에 있으며, CJ프레시웨이는 시니어 전문 식자재 개발 및 공급 서비스를 준비하고 있다.

국내 통신사 SKT는 자사의 AI 스피커 '누구(NUGU)'를 통해 거동이 불편한 어르신들의 음성을 통해 119 호출을 할 수 있는 기능을 선보였다. 또한, 국내 특허 조사에 따르면, 가전제품 내 부착된 버튼을 누르면 연결된 보호자, 긴급구조센터에 위험 상황을 알리는 시스템이 존재한다. 하지만, 이러한 시스템들은 어르신들이 의식을 잃은 상태나, 사망하셨을 때 적절한 기능을 수행하지 못함을 알 수 있다.

2. 낙상 문제

질병관리청에서 제공한 자료에 따르면 우리나라의 경우, 만 65세 이상인 노인의 신체 손상 원인 중 52.2%가 낙상으로 밝혀졌다. 낙상은 집 등 거주 시설에서 발생하는 경우가 48.0%로 가장 많은 것으로 밝혀졌다. 현재는 노인들의 낙상으로 인한 피해에 실시간으로 즉각적으로 대응하기 어려운 형편이다. 보호자나 어르신 도우미들의 도움이 있더라도 시, 공간적 제약 및 물리적 제한이 있어 24시간 종일 관리할 수는 없는 현실이다.

3. 실시간 행동 분석

기존의 연구되었던 실시간 행동 분석을 통한 위험 상황 판단 시스템은 다음과 같은 구조를 가지고 있다.



< RTSP 서버를 이용한 위험 상황 판단 시스템 구조 >

CCTV나 IP 카메라와 연결된 RTSP 서버는 행동인식 서버에 동영상을 전송하고 행동인식 서버를 거쳐며 사전에 학습된 이상행동이 감지되면 감지된 시점의 동영상을 웹 서버에 전송하며, 동영상을 포함한 관련 데이터를 데이터베이스에 저장하고, 클라이언트(보호자)쪽으로 알림 기능을 보내어 사용자가 인지할 수 있게 해준다. 하지만, 이 방식은 몇 가지 문제가 있다.

- 많은 수의 CCTV를 한 번에 제어하게 될 때 성능 저하를 일으킬 수 밖에 없다. 여러개의 동영상들이 프레임 단위로 입력되면, 서버측에 병목 현상이 발생하게 되어 효율적인 작업을 할 수 없게 된다.
- 시간이 지남에 따라, 데이터베이스에 저장되는 데이터의 양이 방대해짐에 따라 이를 관리하기 위한 유지보수 비용이 자동적으로 발생하게 된다. 즉, 기존의 이러한 방식들은 시스템을 이용하는 사람이 많아지면 효율적인 성능을 발휘하기 힘들다.

◇ 개발 과제의 효과

본 과제는 어르신들의 일상 생활 속 행동에 의해 발생하는 위험 상황에 대해 실시간으로 알림을 줄 수 있다. 어르신들의 거주 공간에서 AI 단말기를 사용하는 실시간 행동 분석 시스템을 앱 기반으로 구현하면서, 행동분류 서버를 사용하지 않고 앱 내부에서 처리하여 기존 서비스의 문제였던 성능 저하 문제를 해결할 것이다. 추가적으로 한 기기에 하나의 독자적인 서비스를 제공해 많은 사용자들을 수용할 수 있다.

다. 개발 과제의 목표와 내용

◇ 실시간 영상 분석을 통한 어르신의 행동 분류

- AI 단말기의 내장된 카메라로부터 어르신을 탐지하고, 스켈레톤 구조로 변환한다.
- 스켈레톤 구조로 변환된 특징 벡터를 기반으로 영상의 짧은 시간 안에 어르신들의 행동을 분류한다.
- 분류된 행동이 낙상과 같은 위험 행동으로 분류된 경우 알림을 보낸다.

◇ AI 단말기 내 앱 개발 및 리소스 제어

- AI 단말기의 전원을 키면, 앱이 자동적으로 실행하도록 한다.
- AI 단말기에 내장된 카메라를 제어하여, 앱 내에서 정해진 시간마다 프레임을 가져와 학습된 모델의 입력으로 변환한다.
- AI 단말기 속의 앱 내에서 발생한 알림을 보호자 앱으로 전송한다.

◇ 보호자 앱

- AI 단말기 앱에서 등록한 보호자와 연동되도록 보호자 앱을 구현한다.
- 데이터베이스에서 시니어 정보를 가져와 화면에 출력한다.
- AI 단말기 앱으로부터 전송된 알림을 보호자 앱이 설치된 기기의 상단 알림으로 출력한다.
- 알림을 누르면 시니어의 정보와 스토리지에 저장된 이미지를 화면에 출력한다.

라. 개발과제의 기술적 기대효과

기존의 연구되었던 실시간 행동 분석 관련 프로젝트에서는 CCTV나 IP 카메라에 연결된 RTSP 서버는 행동인식 서버에 실시간으로 동영상을 전송하고, 촬영된 영상은 스마트폰과 같은 사용자의 애플리케이션으로 전송하여 모니터링하는 기법을 사용하고 있다. 하지만, 다수의 CCTV와 연결된 서버는 많은 영상 처리 및 저장과 관련하여 실시간 성능 및 수용력에 한계를 지니며 병목 현상이 이루어지는 문제가 있다. 해당 프로젝트에서는 AI 단말기 내에서 프레임 추출, 행동 분류, 알림 전송 기능을 실행하고 처리하는 형식으로 구현하여 병목 현상을 방지할 수 있을 것이다. 이러한 면에서 많은 사용자들을 수용할 수 있는 수용력을 가진다. 또한 일련의 진행과정에서 서버와의 통신과정이 제외되기 때문에 시간이 단축될 것이다.

마. 개발과제의 경제적 및 사회적 파급효과

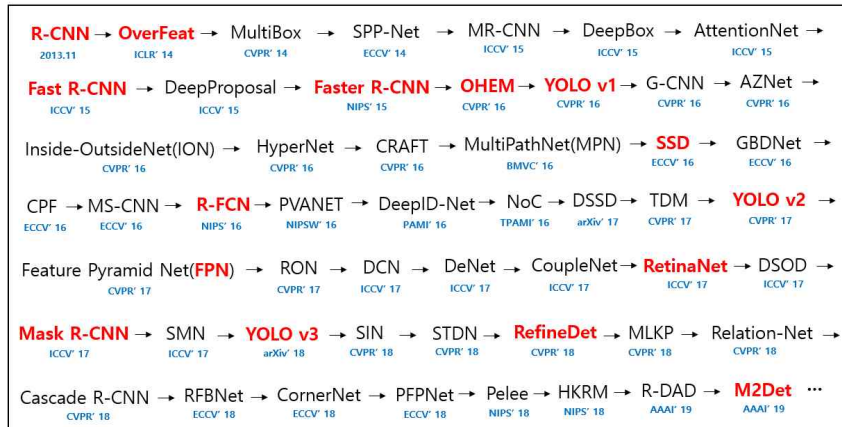
AI 단말기가 시니어들의 주거공간에 보급되고 있는 가운데, 여러 기기들에서 처리하는 것이 아닌, 해당 단말기를 통해 서비스를 제공하므로 추가적인 단말기 구입 비용이 절약된다. 앱 내에서 진행되는 서비스 특성상 서버 구현에 드는 비용도 없다. 또한, 현재 국가 주도 및 지자체에서 어르신들의 건강 관리 및 위험 상황들을 파악 및 예방, 대처하려는 노력을 하고 있다. 이에 관련지어, 일상생활에서 어르신들의 위험상황을 탐지하고 보호자에게 알림을 주어 빠른 대처가 가능할 것이고 한계가 정해져있는 예산 안에서 많은 노인분들을 케어하는 입장인 지자체에게 좋은 선택지가 될 것이다.

1.2 관련 기술의 현황

가. State of art

본 프로젝트의 주요 기술인 Object detection에 대한 기술 현황을 구체적으로 서술한다. 각 기술의 자세한 설명은 본문 분량상 생략한다.

Object Detection: State of the Art Progress

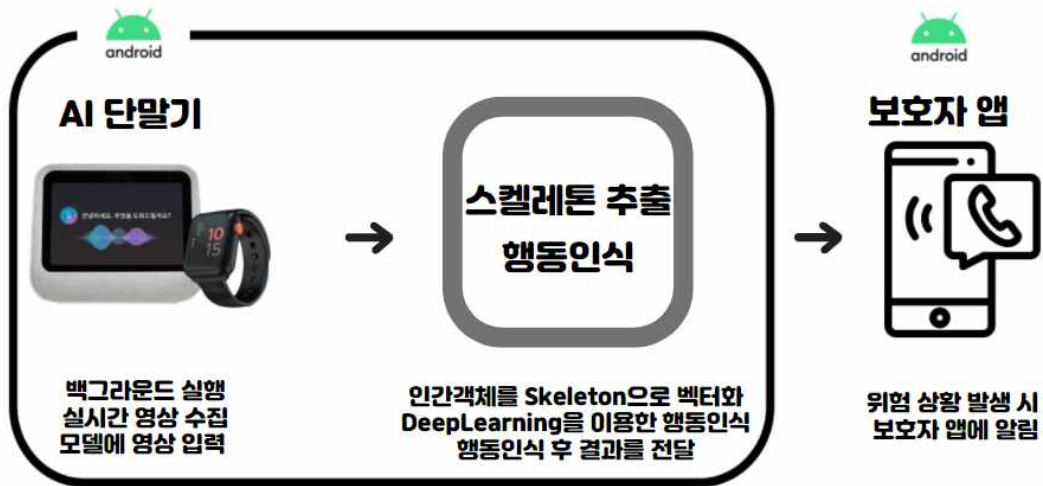


< Object Detection: State of the Art Progress.

source : <https://mlai.ipitek.web.id/2019/01/20/object-detection-state-of-the-art-progress/> >

2013년에 최초의 딥러닝 기반 Object detection 모델인 R-CNN이 등장한 후, R-CNN 계열의 모델인 Fast R-CNN, Faster RCNN 모델들이 성과를 보여주었다. 2016년에는 YOLO와 SSD가 등장하면서 새로운 계열의 딥러닝 기반 모델이 등장하였고, 2017년에 RetinaNet이 등장하면서 높은 성능을 보여주기도 했다. 이러한 모델들이 오랜 시간 동안 각광을 받다가 2019년 후반부터 새로운 모델들이 등장하게 되었다. AutoML 기법을 도입한 EfficientNet, EfficientDet 계열이 등장하였고, NLP 분야에서 사용되는 Attention 기반의 transformer 기법을 도입한 DERT 모델 계열이 등장했다. 또한, 처리 속도는 빠르지만 정확도에서의 한계점을 지닌 YOLO 계열의 모델이 version4, 5를 거듭하여 속도뿐만 아니라 높은 정확도를 가지며 다 시금 객체 탐지 분야에서의 존재를 입증했다. 그 이후에도 성능을 발전시킨 Swin Transformer 기반 모델이 등장하였고, 객체 탐지 분야의 성능 평가인 AP 지표가 60을 넘기는 발전을 기록해오고 있다.

나. 기술 로드맵



< 기술 로드맵 >

AI 단말기 안에 설치할 Android 앱과 알림을 받을 보호자 Android 앱 2가지로 크게 나누어진다. AI 단말기 속 앱은 내장된 카메라를 실시간으로 제어하여 프레임을 추출하게 된다. 추출된 프레임을 미리 학습한 모델의 입력으로 사용한다. 이때, 모델에는 프레임 속에 존재하는 인간을 스켈레톤으로 벡터화하고, 이 벡터화된 데이터를 딥러닝 모델을 이용하여 어떤 행동을 취하고 있는지 분류하게 된다. 딥러닝 모델이 사전에 정해진 위험상황을 결과로 출력하면, 보호자 앱에 알림을 주는 기능을 통해 위험 상황을 대처하도록 설계했다.

다. 특허조사

1. 독거노인 케어 시스템 (법정상태 : 거절, 심사진행상태 : 거절결정(재심사))

급작스러운 신체적 문제 발생에 따라 독거노인이 스스로 자신의 상황을 전달할 수 없는 형편인 경우에도 독거노인의 이상징후를 스스로 판단하여 이상 결정 시 이를 관리주체에 통보할 수 있는 독거노인 케어 시스템을 제공하려는 것으로서, 본 발명에 따른 독거노인 케어 시스템은, 노인이 거주하는 세대 내에 설치되는 전기기기 스위치의 온(ON)/오프(OFF) 상태정보를 감시하고, 온/오프 상태정보와 전기기기 스위치의 신원정보를 함께 제공하는 통신부를 갖는 세대별 감시 통신장치이다.

2. 리모콘을 이용한 위험 관리 시스템 (공개특허공보)

리모콘을 이용한 위험 관리 시스템이 개시된다. 본 발명의 일 실시예에 따른 리모콘을 이용한 위험 관리 시스템은 버튼의 입력정보를 전송하는 TV리모콘 및 상기 TV리모콘과 무선으로 연결되어 있으며, 상기 TV리모콘으로부터 상기 입력정보를 수신하고, 상기 스마트폰의 입력정보를 분석하여 상기 TV리모콘 사용자의 신변에 변화가 발생한 것으로 판단되는 경우 지정된 연락처로 통보하는 스마트폰을 포함한다.

1.3 관련 시장에 대한 분석

가. 경쟁제품 조사 비교

실시간 행동 분석 및 객체 탐지 기술은 여러 분야에 있어 보급되고 있으며, 더 높은 성능 향상을 위한 연구가 계속되서 진행되고 있다. 일례로, 신한은행은 보이스피싱의 타깃이 된 고령 고객의 금융자산을 보호하기 위한 특단의 대책으로 은행권 처음으로 'AI 이상행동 탐지 ATM'을 도입한다고 2022년 3월 7일에 밝혀왔다. 또한, SKT는 AI 기반 지능형 영상분석 솔루션을 상반기 출시할 예정이며, 자체 AI칩으로 개발을 할 계획을 내비추었다. 해당 기술은 주로 공공 안전, 국방 경계, 산업 안전 등에 활용될 예정이다. 이미 상용화된 기술도 있다. 교통 분야에서는 2020년 '주차유도' 기술을 상용화했다. 영상 내 주차 공간을 인식해 사용자에게 주차 자리 빈 곳을 알려주고, 안정적인 주차를 유도하는 기술이다. 또, 운전자 모습을 영상으로 실시간 촬영해 위험 여부를 알려주는 '안전운전 보조 솔루션'도 선보였다.

앞선 딥러닝 기술들은 어르신들을 위한 서비스 제공에도 이루어지고 있다. 노년층을 위한 '시니어 맞춤형 서비스'가 다양한 산업군에서 쏟아지고 있다. 한글과컴퓨터는 시니어 케어 서비스 사업에 진출하여 '한컴 말랑말랑 행복케어'를 통해 인지훈련 치매 예방 가상현실(VR)과 상호교감이 가능한 AI 로봇 활용 프로그램도 제공할 예정이라고 밝혔다. 또, SKT는 자사의 AI 스피커인 '누구(NUGU)'를 통해 시니어 케어 서비스를 제공하고 있다. 이를 통해 거동이 불편한 어르신들이 음성을 통해 119 호출을 할 수 있는 기능과 말동무 기능 등을 선보였다. 또 다른 이동통신사 KT와 LGU+에서도 시니어 케어 사업에 적극적인 태도를 취하고 있는 것으로 나타났다.

나. 마케팅 전략

질병관리청에서 제공한 자료에 따르면 우리나라의 경우, 만 65세 이상인 노인의 신체 손상 원인 중 52.2%가 낙상으로 밝혀졌다. 낙상은 집 등 거주 시설에서 발생하는 경우가 48.0%로 가장 많은 것으로 밝혀졌다. 현재는 노인들의 낙상으로 인한 피해에 즉각적으로 대응하기 어려운 형편이다. 보호자나 도우미들의 도움이 있더라도 시, 공간적 제약 및 물리적 제한이 있어 24시간 종일 관리할 수는 없다.

따라서, 본 프로젝트는 기존의 디스플레이형 AI 단말기에 있는 카메라를 통해 실시간으로 어르신들의 위험 상황을 판별하고, 이를 보호자에게 알람을 주는 시스템(앱)을 개발하여 사회문제를 해결하고자 한다. 여러 장비가 필요하지 않은 점과 행동분류모델이 빠르다는 점은 어르신들을 돌보는 입장인 보호자, 도우미, 지자체 쪽에 좋은 선택지가 될 것이고 이쪽을 중점으로 마케팅을 진행할 것이다.

2. 설계

2.1 사용자 요구사항

번호	요 구 사 항	D or W	비고
1	죽지 않는 서비스를 구현한다.	D	中
2	단말기 카메라를 이용하여 매초마다 일정 개수의 이미지를 수집한다.	D	中
3	보호자 정보를 설정한다.	D	小
4	위험 상황 판단 시, 보호자에게 알림을 보낸다.	D	大
5	학습모델을 단말기 내에서 실행될 수 있도록 이식한다.	D	大
6	영상처리를 통해 인간 객체를 스켈레톤화한다.	D	大
7	스켈레톤화된 데이터를 딥러닝을 가지고 행동분류한다.	D	大
8	학습 모델의 처리 속도가 10초를 넘기지 않는다.	D	大
9	카메라가 비추지 못하는 구역을 소음분류를 통해 판단한다.	W	小
10	이미지를 주기적으로 저장/삭제한다.	W	小
11	단말기에서 보호자 측으로 보내는 알림이 5초 안에 송신된다.	W	中
12	분류해야하는 행동의 종류를 5개 이상으로 한다.	W	小

2.2 사용자 요구사항 만족을 위한 기능 정의 및 기능별 정량목표

- ◇ 테스트 행동에 대한 분류
 - 분류 정확도 : 전,후,좌,우,예외를 포함하여 90% 이상
- ◇ 행동 분류 모델 처리
 - 속도 : 평균 10초 미만 (입력부터 출력까지)
- ◇ 알림 메시지 전송
 - 속도 : 평균 5초 미만 (모델 출력값이 나온 후부터)
- ◇ 이미지 프레임 수집
 - 속도: 평균 0.5초마다 수집

2.3 기능 구현을 위한 세부기술 선택사항 (디자인)

가. 시스템 구성

◇ 파이어베이스(실시간 데이터베이스)

- 실시간으로 데이터를 저장하고 사용자 간에 동기화할 수 있는 클라우드 호스팅 NoSQL 데이터베이스
- NoSQL이기 때문에 앱이 필요로 하는 형태의 데이터로 저장/수정이 유연하게 가능
- 사용자는 실시간 동기화를 통해 웹이나 모바일에서 손쉽게 기기의 데이터에 액세스가능
- 모바일 및 웹 SDK와 함께 제공되므로 서버 없이도 앱을 빌드가능

나. 통신 및 인증

◇ 파이어베이스(FCM)

- 무료로 메시지를 안정적으로 전송할 수 있는 교차 플랫폼 메시징 솔루션
- FCM의 신뢰성 높고 배터리효율이 좋은 연결 채널을 통해 기기에서 기기로 메시지를 보낼 수 있음
- 회원가입 할 때 얻은 등록토큰을 가지고 메시지를 보내는데 등록토큰은 앱을 설치한 기기마다 고유한 값

◇ 파이어베이스(인증)

- 사용자 인증 시 필요한 백엔드 서비스와 사용하기 쉬운 SDK, 기성 UI 라이브러리를 제공
- 이메일 및 비밀번호 기반 인증이지만 약간의 커스텀을 해서 아이디 및 비밀번호 기반 인증 형태로 변경
- 인증을 마친 사용자가 파이어베이스 실시간 데이터베이스와 클라우드 스토리지에서 데이터 write/read 가능

다. 앱 개발 및 디버깅 모델

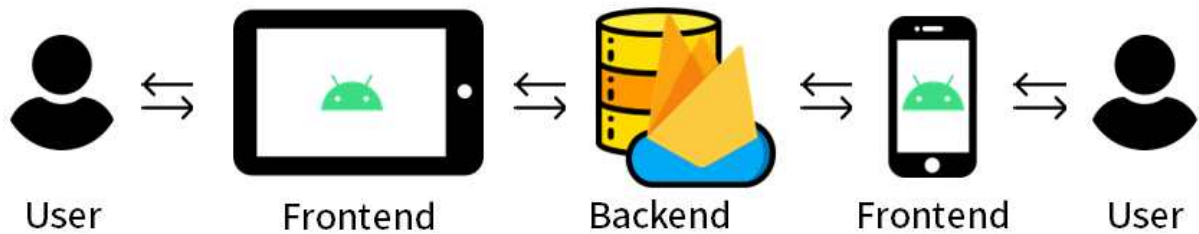
◇ surfaceview

- View를 상속받는 클래스
- 스레드를 이용해 강제로 화면에 그리는 행위를 실행함으로써 원하는 시점에 바로 화면에 그릴 수 있음
- 애니메이션이나 동영상과 같이 연산 처리가 많이 필요한 뷰를 위해 사용됨
- 더블 버퍼링 기법을 이용하여 SurfaceHolder가 Surface에 미리 그리고 이 Surface가 SurfaceView에 반영되는 방식
- 자기 영역 부분의 Window를 뚫어서 자신이 보여지게끔 하고 Window와 View가 블렌딩되어 화면에 보여짐

◇ 텐서플로 라이트

- 개발자가 모바일, 내장형 기기, IoT 기기에서 모델을 실행할 수 있도록 지원하여 기기 내 머신러닝을 사용할 수 있도록 하는 도구 모음
- 기기 내 머신러닝에 최적화
- 지연 시간(서버까지의 왕복 없음), 개인 정보 보호(기기에 개인 정보를 남기지 않음), 연결성(인터넷 연결이 필요하지 않음), 크기(모델 및 바이너리 크기 축소), 전력 소비(효율적인 추론 및 네트워크 연결 불필요)에 대한 제약사항 해결
- 하드웨어 가속 및 모델 최적화 사용
- TensorFlow 모델을 TensorFlow Lite모델로 변환하여 기기에 이식

2.4 시스템 설계



◇ Frontend

- Android Studio를 이용하여 구현한다.
- 회원가입, 로그인, 주요 기능을 위한 화면을 생성하고, User가 발생시키는 이벤트를 받아 Backend단으로 넘긴다.

◇ Backend

- Firebase가 제공하는 기능들을 이용하여 구현한다.
- 실시간 데이터베이스 기능을 이용하여 User들의 정보를 저장/수정한다.
- 인증 기능을 이용하여 User 인증을 진행한다.
- 스토리지 기능을 이용하여 이미지를 저장/불러오기한다.
- 호스팅 기능을 이용하여 주소API를 가져와 주소검색 액티비티를 제공한다.
- Frontend단에서 온 이벤트를 처리하고 결과값을 형태에 맞춰 반환한다.
- 덤퍼닝 모델은 서버에서 처리하지 않고 앱 내부에서 처리하여 결과만 Backend로 넘긴다.

◇ 장점

- 모든 데이터는 Backend에 모이기 때문에 데이터의 구성과 관리 측면에서 유용하다.
- 서버를 따로 만들어서 쓰는 것이 아니라 제공하는 서비스를 통해 쓰기 때문에 유지보수에 좋고 편리성이 뛰어나다.

2.5 이론적 계산 및 시뮬레이션

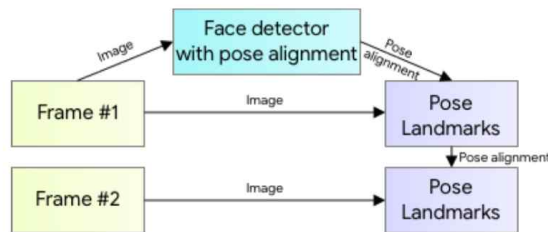
가. Pose Detection API의 이론적 과정

Pose Detection API는 BlazePose를 기반으로 작동하는 API이다.

BlazePose는 모바일에서도 human pose estimation에 대한 real-time inference가 가능한 모델이다. BlazePose는 heatmap과 regression을 모두 이용해 lightweight pose estimation을 한다는 장점이 있다.

기존의 자세 추정 모델은 heatmap을 이용해 각 관절에 대한 heatmap을 생성해 각 관절 좌표에 대한 offset을 수정하는 방식으로 진행되었지만, single person에 대한 실시간 자세 추론을 하기에는 모델이 너무 크다는 단점이 존재했다.

Regression 기반 모델은 컴퓨팅 리소스가 저렴하고 확장성이 크나, 관절이 맞물리는 경우에 예측이 취약하다는 단점이 있지만, BlazePose는 인코더 디코더 신경망을 이용해 모든 관절의 heatmap을 예측한 후, 다른 encoder로 regression을 이용해 각 관절의 좌표를 예측하는 구조를 가지고있으며, 추론당시에 heatmap부분을 제외해 모바일에서도 돌릴 수 있을 정도로 가볍게 만들어진다.

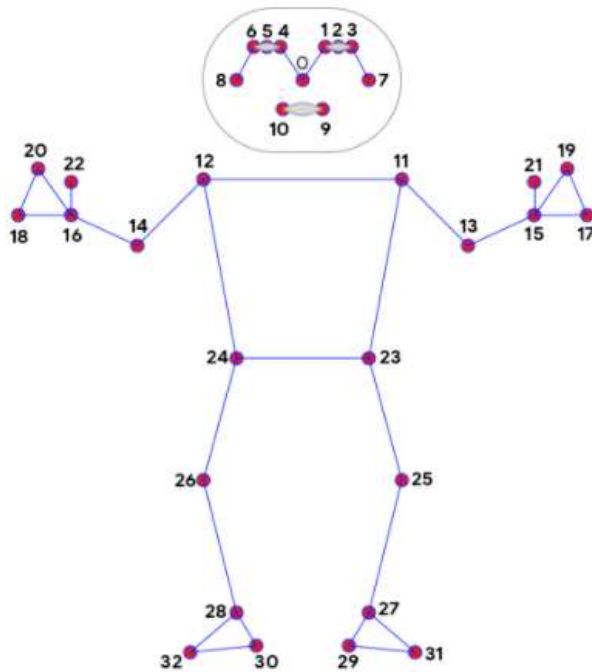
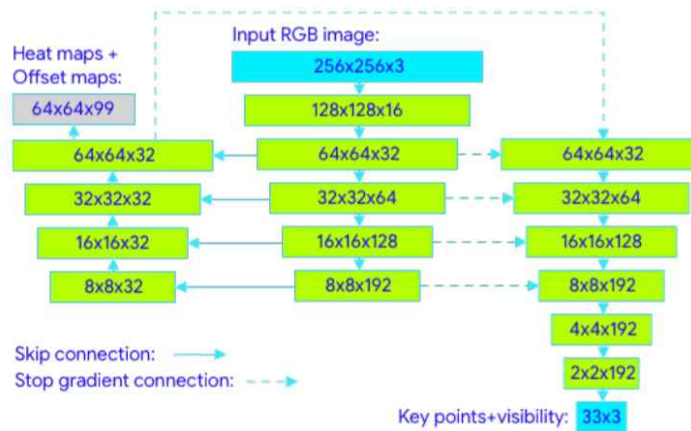


추론 파이프라인은 lightweight body pose detector와 pose tracker 순서로 구성된다. tracker는 keypoint 좌표, 사람 존재 유무, 현재 프레임의 ROI(Region Of Interesting)를 예측한다. Tracker에서 사람이 없다고 판단할 경우 다음 프레임에서 detector를 재실행한다.

최근의 객체 탐지 방법은 대부분 NMS(Non-Maximum Suppression)으로 후처리를 한다. 하지만 NMS는 keypoint가 겹치는 동작에서 오류가 발생한다는 단점이 있다.

BlazePose에서는 detector의 기준을 명확한 특징이있고, 편차가 적은 얼굴로 정하여, 사람의 골반 중앙 값, 사람을 포함하는 원의 크기, 사람의 기울기 등의 alignment parameter를 예측한다.

BlazePose는 heatmap, offset, regression을 결합한 방식을 이용했다. Heatmap과 offset Loss는 학습 과정에서만 이용하고, 추론을 할 때에는, output Layer를 제거해 경량화하여 regression encoder에서 활용할 수 있도록 했다. 해당 방식은 인코더-디코더 히트맵 기반 신경망 뒤에 regression 신경망이 따라오는 구조를 쌓는 방식으로 적용했다.



- | | |
|--------------------|----------------------|
| 0. nose | 17. left_pinky |
| 1. left_eye_inner | 18. right_pinky |
| 2. left_eye | 19. left_index |
| 3. left_eye_outer | 20. right_index |
| 4. right_eye_inner | 21. left_thumb |
| 5. right_eye | 22. right_thumb |
| 6. right_eye_outer | 23. left_hip |
| 7. left_ear | 24. right_hip |
| 8. right_ear | 25. left_knee |
| 9. mouth_left | 26. right_knee |
| 10. mouth_right | 27. left_ankle |
| 11. left_shoulder | 28. right_ankle |
| 12. right_shoulder | 29. left_heel |
| 13. left_elbow | 30. right_heel |
| 14. right_elbow | 31. left_foot_index |
| 15. left_wrist | 32. right_foot_index |
| 16. right_wrist | |

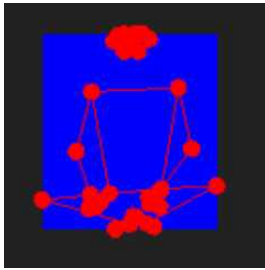
< Pose Detection API의 출력으로 나오는 33개 관절 정보 >

나. 1차 판단 알고리즘 이론적 과정 및 시뮬레이션

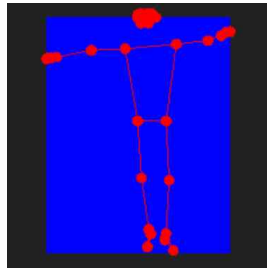
기존에 낙상에 따른 신체 관절의 동적 특성을 분석한 논문[1]에 따르면, 낙상 행동에서 땅에 닿기까지 상체는 약 0.8초, 하체는 약 0.5초의 시간이 걸린다고 분석했다. 하지만 해당 1차 판단 알고리즘에서는 어르신들의 움직임을 고려해서 시간의 폭을 좀 더 늘려 사람이 서 있던 자세에서 1초만에 바닥에 쓰러진 자세로 인식된다면 이를 낙상으로 의심한다. 이 특성과 Pose Detection API의 관절 정보로 그린 바운딩 박스의 비율로 1차적으로 낙상 의심 상황을 검출한다.

◇ 바운딩 박스

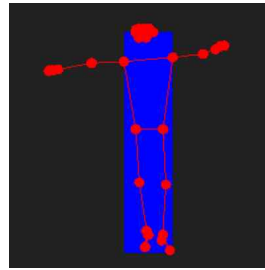
스켈레톤으로 낙상을 감지했던 기존 논문[4]을 참고하여 바운딩 박스를 이용하여 1차 판단을 사용하기로 하였다. 하지만 기존 알고리즘과 해당 알고리즘의 차이점은 Pose Detection API 추출한 33개의 정보 중에서 팔과 관련된 관절 정보(팔꿈치, 손목, 손가락)를 제외한 나머지 관절들을 사용하여 바운딩 박스를 그린다는 점이다. 팔을 제외한 이유는 서 있을 때의 자세와 앉아있을 때의 자세를 구별하기 위함이다. 일반적으로 사람이 바닥에 앉을 때는 바운딩 박스가 정사각형 모양이고, 서 있는 경우에는 세로로 긴 직사각형 모양이다. 이런 상황이라면 서 있을 때나 앉아있을 때를 구별할 수 있다. 하지만, 만약 사람이 서 있는 상태로 양팔을 벌린다면, 아래의 설명 그림 1, 2처럼 두 자세의 바운딩 박스는 둘 다 정사각형에 가깝게 그려지기 때문에 바운딩 박스만 보고 서 있거나 앉아있는 것은 구분하는 것은 어렵다. 이외에도 어떤 자세를 취하든 팔은 움직임이 많기 때문에, 팔을 제외하지 않고는 바운딩 박스의 수식 조건을 일반화하는 것은 어려웠기 때문에 바운딩 박스를 그릴 때 양팔은 제외하기로 하였다. 그 결과, 그림 3, 4처럼 사람이 서 있는 상태로 팔로 다른 행동을 하고 있더라도 바운딩 박스가 똑같이 그려지기 때문에 서 있다는 상태를 판단할 수 있게 되었다.



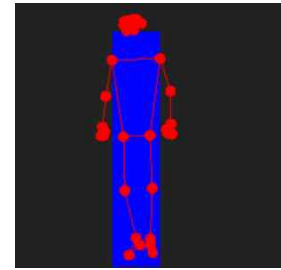
< 설명 그림1
앉을 때 (양팔 포함) >



< 설명 그림2
팔을 벌리고 서 있을 때 (양팔 포함) >



< 설명 그림3
팔을 벌리고 서 있을 때 (양팔 제외) >



< 설명 그림4
서 있을 때 (양팔 제외) >

이렇게 구한 바운딩 박스의 폭과 높이를 가지고, 바운딩 박스의 비율 R 을 계산한다. 일반적으로 서 있거나 앉아서 무언가를 한다면 $R < 1$, 바닥에 길게 누워있는 상태에서는 $R > 1$ 이다.

$$R = \text{Width} / \text{Height}$$

◇ 1차 판단 조건문

앞선 바운딩 박스 비율로 현재의 자세를 판단하고, 그 자세가 1.0초만에 서 있는 자세에서 누워있는 자세로 변경된다면 이를 낙상 의심 상황으로 판단한다. 그러기 위해서는 일단 일정 시간마다 바운딩 박스의 비율을 계산해야 한다. $w(t)$, $h(t)$ 는 각각 현재 시간 t 의 바운딩 박스의 폭과 높이이고, $R(t)$ 는 현재 시간의 바운딩 박스의 비율이다.

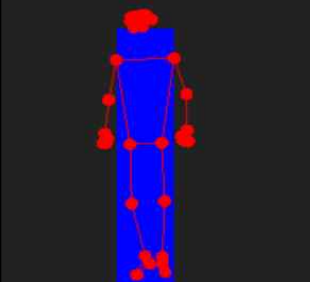
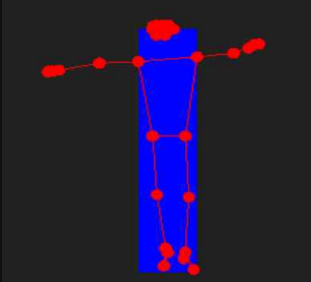
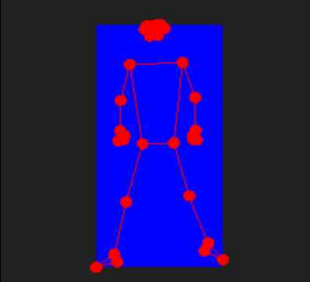
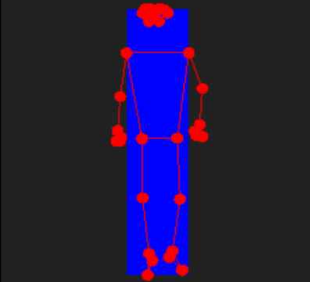
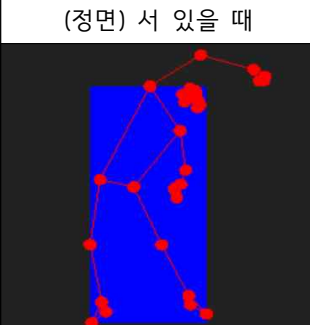
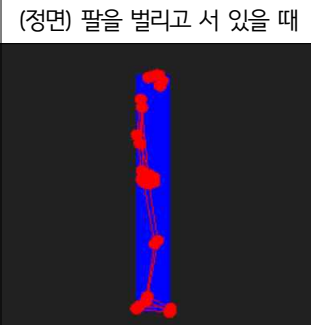
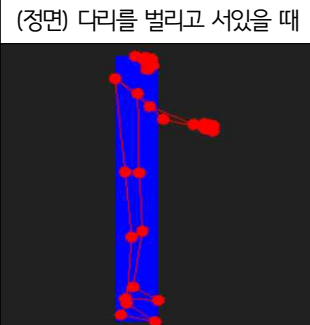
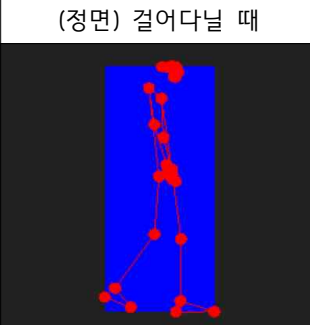
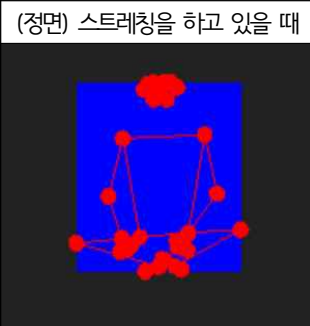
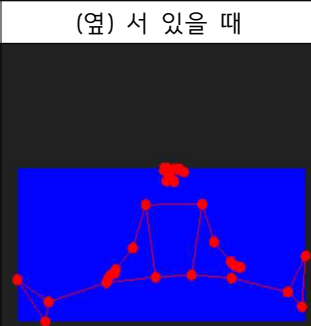
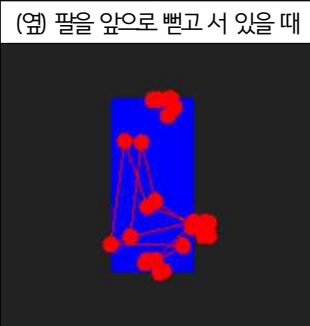
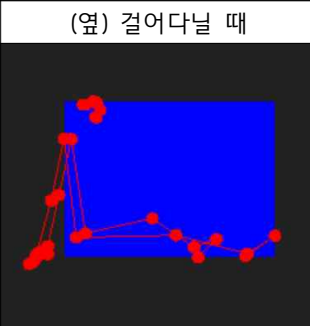
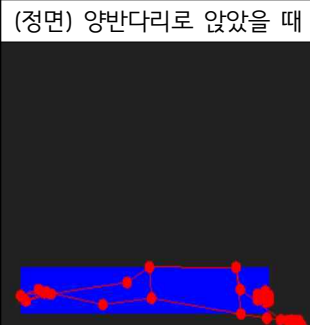

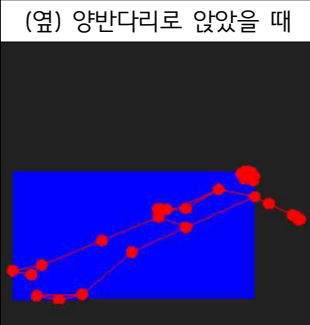

$$R(t) = \frac{w(t)}{h(t)}$$

현재 시간의 $R(t)$ 값과 1.0초전의 바운딩 박스의 비율 $R(t-2)$ 값을 사용하여 조건식을 만들면 1.0초만에 서 있던 자세가 누운 자세로 변경되었는지를 판단할 수 있게 된다. 아래의 조건식에서 α , β 값은 각각 서 있는 자세를 구분하기 위한 임계값, 쓰러진 자세를 구분하기 위한 임계값이다.

$$\text{if } R(t-2) < \alpha \text{ and } R(t) > \beta : \\ \text{낙상 의심}$$

◇ 임계값을 찾기 위한 시뮬레이션

바운딩 박스의 비율로 자세를 구분할 수 있는 적절한 임계값을 구하기 위해서 일상생활에서 일어날 수 있는 자세들을 직접 취하고, 해당 자세의 비율을 분석하였다. 정면이란 카메라를 정면으로 바라봤을 때를 의미하고, 옆은 카메라가 찍는 방향에 수직한 방향을 바라봤을 때를 의미한다.

			
(정면) 서 있을 때	(정면) 팔을 벌리고 서 있을 때	(정면) 다리를 벌리고 서있을 때	(정면) 걸어다닐 때
			
(정면) 스트레칭을 하고 있을 때	(옆) 서 있을 때	(옆) 팔을 앞으로 뻗고 서 있을 때	(옆) 걸어다닐 때
			
(정면) 양반다리로 앉았을 때	(정면) 다리를 벌리고 앉았을 때	(옆) 양반다리로 앉았을 때	(옆) 다리를 뻗고 앉았을 때
			
쓰러졌을 때, 누울 때	카메라에 머리를 두고 쓰러졌을 때	카메라에 발을 두고 쓰러졌을 때	대각선으로 쓰러졌을 때,

< 여러 자세들의 바운딩 박스 >

실험한 자세 바운딩 박스의 R(t)값의 범위를 측정한 결과는 다음과 같다.

번호	자세	R(t)
1	(정면) 서 있을 때	0.18~0.21
2	(정면) 팔을 벌리고 서 있을 때	0.18~0.21
3	(정면) 다리를 벌리고 서있을 때	0.49~1.05
4	(정면) 걸어다닐 때	0.20~0.24
5	(정면) 스트레칭을 하고 있을 때	0.34~0.43
6	(옆) 서 있을 때	0.11~0.15
7	(옆) 팔을 앞으로 뻗고 서 있을 때	0.11~0.21
8	(옆) 걸어다닐 때	0.37~0.47
9	(정면) 양반다리로 앉았을 때	0.78~0.89
10	(정면) 다리를 벌리고 앉았을 때	1.80~2.04
11	(옆) 양반다리로 앉았을 때	0.49~0.68
12	(옆) 다리를 뻗고 앉았을 때	1.31~1.80
13	쓰러졌을 때, 누울 때	5.0 이상
14	카메라에 머리를 두고 쓰러졌을 때	1.46 이상
15	카메라에 발을 두고 쓰러졌을 때	1.2 이상
16	대각선으로 쓰러졌을 때,	3.0 이상

팔을 제외하였기 때문에 일반적으로 다리를 벌리고 서있는 경우를 제외하면 서 있거나 걸어 다닐 때 (1,2,4,5,6,7,8) 0.45이하의 R(t) 값을 가진다. 또한, 쓰러진 자세(13,14,15,16)는 찍히는 각도에 따라 값의 변동이 심했지만 누워있다는 특징으로 인해 $w(t)$ 의 값이 $h(t)$ 의 값보다 일반적으로 비율이 컸다. 쓰러진 자세 중에서도 $w(t)$ 가 가장 짧게 찍히는 각도인 14,15의 경우에도 R(t)값이 대략 1.2보다는 크다는 것을 알 수 있었다. 이 결과를 통해서 서 있는 자세를 구분하는 임계값 α 값은 0.45, 쓰러진 자세를 구분하는 임계값 β 값은 1.2로 정하였다. β 인 1.2를 넘는 자세에는 쓰러지기 자세 외에도 10,12번 등의 자세들이 있지만, 해당 항목들은 서 있다가 다른 중간 동작없이 1.0초만에 취할 수 없는 자세들이다. 임계값을 이렇게 정함으로써, 서 있다가 앉는 경우나 앉아있다가 눕는 경우는 임계값으로 세워진 조건식을 만족하지 못하기 때문에 서 있다가 쓰러지는 경우만 잡을 수 있게 된다.

◇ 프레임 속도

해당 알고리즘에서는 1.0초를 기준으로 자세를 판단한다. 하지만 1.0초 간격으로 촬영한다면 $t-2$ 초와 t 초 사이(1.0초)인 중간 지점부터 넘어지는 자세가 시작한다면 정확도가 떨어지게 된다. 그렇기 때문에 이 간격을 줄이고자 프레임 속도를 0.5초로 하여, 0.5초마다 자세를 추출하고 대신 1.0초 전의 값을 비교하는 알고리즘에는 영향을 받지 않도록 2 프레임 전의 R(t)값을 사용하는 방법을 사용한다.

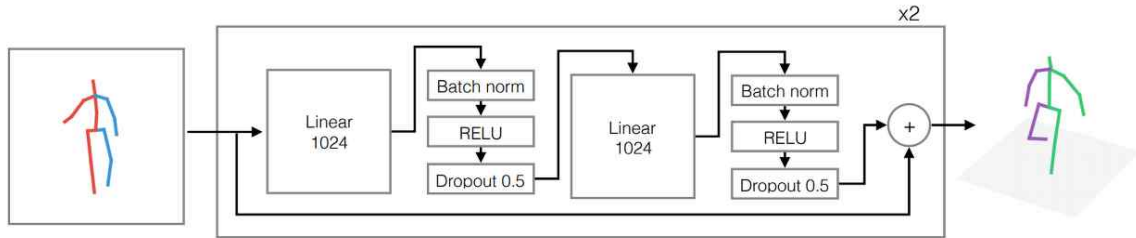
◇ 조건식

앞선 내용을 모두 종합하여 1차적으로 낙상 의심 상황을 판단하는 조건식을 다음과 같다.

if $R(t-2) < 0.45$ and $R(t) > 1.2$:
낙상 의심

$R(t) = \frac{w(t)}{h(t)} = \frac{\text{바운딩 박스의 가로 길이}}{\text{바운딩 박스의 세로 길이}}$
 $R(t-1) = 1$ 프레임 (0.5초전)전의 R(t) 값
 $R(t-2) = 2$ 프레임 (1.0초전)전의 R(t) 값

다. 3D-Pose-Baseline의 이론적 과정



3D-Pose-Baseline은 딥러닝 기반 모델로 주어진 2D 관절 좌표를 3차원으로 변환해주는 모델이다.

$$f^* = \min_f \frac{1}{N} \sum_{i=1}^N L(f(x_i) - y_i)$$

(N : Number of Action, i : Number of Articulation)

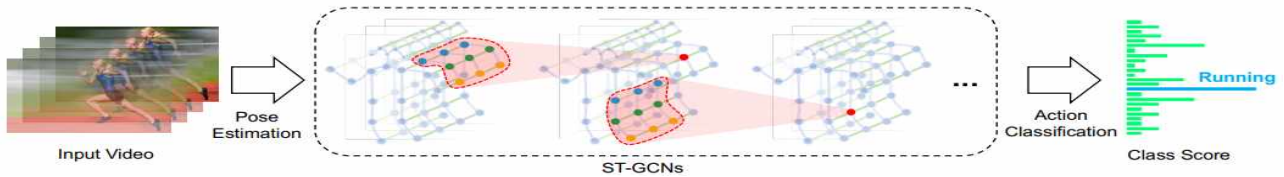
기본적인 수식은 위의 수식과 같이 2차원의 점 $x \in R^{2n}$ 을 딥러닝 모형 $f^* : R^{2n} \rightarrow R^{3n}$ 을 통해 예측된 값 $f(x)$ 와 실제 타깃값(실측값)인 점 $y \in R^{3n}$ 과의 오차가 최소화되도록 하는 함수를 학습하도록 한다.

즉, x_i 는 카메라 parameter들을 알고 있는 상태에서 얻은 2D Ground Truth이거나 2D 자세추정 알고리즘에서 얻은 좌표값이다. 최종적으로 얻은 3차원의 좌표들은 global position이 아닌 hip 관절을 원점 중심으로 해서 이루어져있다.

네트워크의 Building Block은 배치정규화와 드롭아웃, ReLU 활성화함수를 포함하는 선형 레이어고, Residual Connection으로 둘러싸인 2개의 블록으로 2번 반복된다.

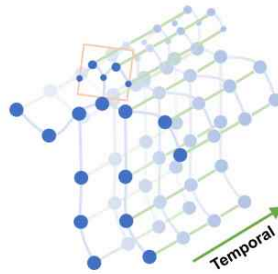
라. ST-GCN의 이론적 과정

행동 인식을 하기 위해 사용하는 데이터셋의 modality는 다양하다. RGB 영상, depth 영상, 움직임의 방향이 표현된 optical flow, 그리고 skeleton data가 있다. ST-GCN은 skeleton을 이용한 행동인식 분류 모델이다. skeleton data의 경우 자연스럽게 사람의 관절의 위치를 시간축에 따라 표현되어진다. 이러한 구조의 data로 학습을 진행하기 위해 이전의 방식에서는 한 frame에서의 관절의 위치 값들을 하나의 feature vector로 사용해서 학습을 시켰었다. 이 방식의 경우, 행동인식에서 관절 간의 관계성 또한 매우 중요함에도 불구하고 해당 정보가 학습에 포함되지않는다. 해당 모델은 공간적 관계성을 찾아내기 위해 GCN(Graph Convolutional Neural Network)를 이용한다.



개략적인 모델의 개요는 위의 그림과 같다. 동영상으로부터 skeleton을 추출하고, skeleton data를 그래프 형태로 만든다. 각각의 joint들을 노드로 만들고, 각각의 노드가 이어지는 부분(공간, 시간)을 edge로 연결한다. 총 9개의 ST-GCN 모듈을 통해 feature를 추출한 후, Softmax 함수를 이용하여 행동을 분류한다.

◇ Skeleton Graph Construction



위의 그림과 같이 Skeleton data를 그래프로 만든다. 모든 frame에서의 skeleton의 joint들이 하나하나의 vertex로 구성되고, 한 frame 내에서의 주변 joint들과 edge로 연결된다. 또한 frame 사이에서 같은 joint끼리도 edge로 연결된다. 이러한 그래프 구조는 dataset의 종류와 관계없이 모두 동일하게 적용 가능하다.

그래프에서 vertex의 집합은 다음과 같이 표기된다 T 는 frame의 개수, N 은 한 Skeleton에서 joint의 개수를 의미한다.

$$V = \{v_{ti} | t = 1, \dots, T, i = 1, \dots, N\}$$

edge의 경우 두 종류로 나뉘는데 한 frame안에서 joint의 edge는 다음과 같다.

$$E_S = \{v_{ti}v_{tj} | (i, j) \in H\}, \quad H \text{ is the set of naturally connected human body joints}$$

$$E_T = \{v_{ti}v_{(t+1)i}\}$$

◇ Graph Convolutional Neural Network

ST-GCN을 설명하기 전, GCN에 대해 먼저 설명하겠다. 한 장의 프레임을 가지는 GCN은 다음과 같은 수식을 가진다. 시점 τ 에서의 한 프레임에서 N 개의 joint node V_t 와 edge $E_S(\tau) = \{v_{ti}v_{tj} | t = \tau, (i, j) \in H\}$ 를 가질 것이다. 이때, kernel size가 $K \times K$ 이며, 채널의 수 c 를 가지는 input 특성맵 f_{in} 이 주어졌다고 하자. 공간위치 x 에서의 단일 채널의 output 값은 다음과 같이 쓸 수 있다.

$$f_{out}(x) = \sum_{h=1}^K \sum_{w=1}^K f_{in}(p(x, h, w)) \cdot w(h, w)$$

여기서, sampling function $p : Z^2 \times Z^2 \rightarrow Z^2$ 한 위치 x 로부터 주변 픽셀들을 가져오는 함수이다. weight function $w : Z^2 \rightarrow R^c$ 는 위의 sampling function에서 얻어와진 픽셀들에 weight를 더하는 함수이다.

◇ Sampling function

이미지에서는 sampling function $p(h, w)$ 는 중심 위치 x 에 대하여 이웃하는 픽셀들이라고 정의한다. 그래프에서는 이미지에서와 유사하게 정의하며 다음과 같은 이웃 집합 B 를 정의한다.

$$B(v_{ti}) = \{v_{tj} | d(v_{tj}, v_{ti}) \leq D\} \text{ of a node } v_{ti}$$

여기서 $d(v_{tj}, v_{ti})$ 는 v_{tj} 로부터 v_{ti} 로 도달하는 어떠한 길들 중에 가장 짧은 길이를 의미한다. 여기서 D 는 이웃을 정의하는 거리의 threshold라고 생각할 수 있다.

정리하면, 그래프에서의 sampling function $p : B(v_{ti}) \rightarrow V$ 는 다음과 같이 쓸 수 있다.

$$p(v_{ti}, v_{tj}) = v_{tj}$$

◇ Weight function

이미지의 경우 중심 위치 주위로 이미 정해진 grid가 존재하기 때문에, 거리적으로 weight가 정해져있다. 반면 그래프의 경우 어떤 것이 더 가까운지 정의하기 어렵기 때문에, 해당 모델에서는 모든 이웃 노드들에게 unique label을 주는 것 대신에, 특정 joint node v_{ti} 의 이웃 집합 $B(v_{ti})$ 을 고정된 K 개의 부분 집합으로 분할하는 방법을 선택한다. 각 부분 집합은 숫자 label을 가지고 있다.

그러므로, 이웃 노드들을 부분 라벨 집합으로 넣는 매핑함수 $l_{ti} : B(v_{ti}) \rightarrow \{0, \dots, K-1\}$ 가 존재한다.

결국, weight function $w(v_{ti}, v_{tj}) : B(v_{ti}) \rightarrow R^c$ 는 (c, K) 차원 tensor를 인덱싱하여 실행되거나, 다음과 같이 정의된다.

$$w(v_{ti}, v_{tj}) = w(l_{ti}(v_{tj}))$$

◇ Spatial Graph Convolution

앞서 설명한 이미지에서의 GCN을 그래프 형태로 다시 쓰면 다음과 같다.

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(p(v_{ti}, v_{tj})) \cdot w(v_{ti}, v_{tj})$$

위의 식에서 $Z_{ti}(v_{tj}) = |\{v_{tk} | l_{ti}(v_{tk}) = l_{ti}(v_{tj})\}|$ 는 대응하는 부분집합의 기수와 동일하다. 해당 항은 출력에 다른 부분 집합들의 기여도를 균형 잡기 위해 넣었다. 앞서 정의한 sampling function과 weight function을 위의 식에 대입하면 최종적으로 다음과 같은 식을 만족한다.

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot w(l_{ti}(v_{tj}))$$

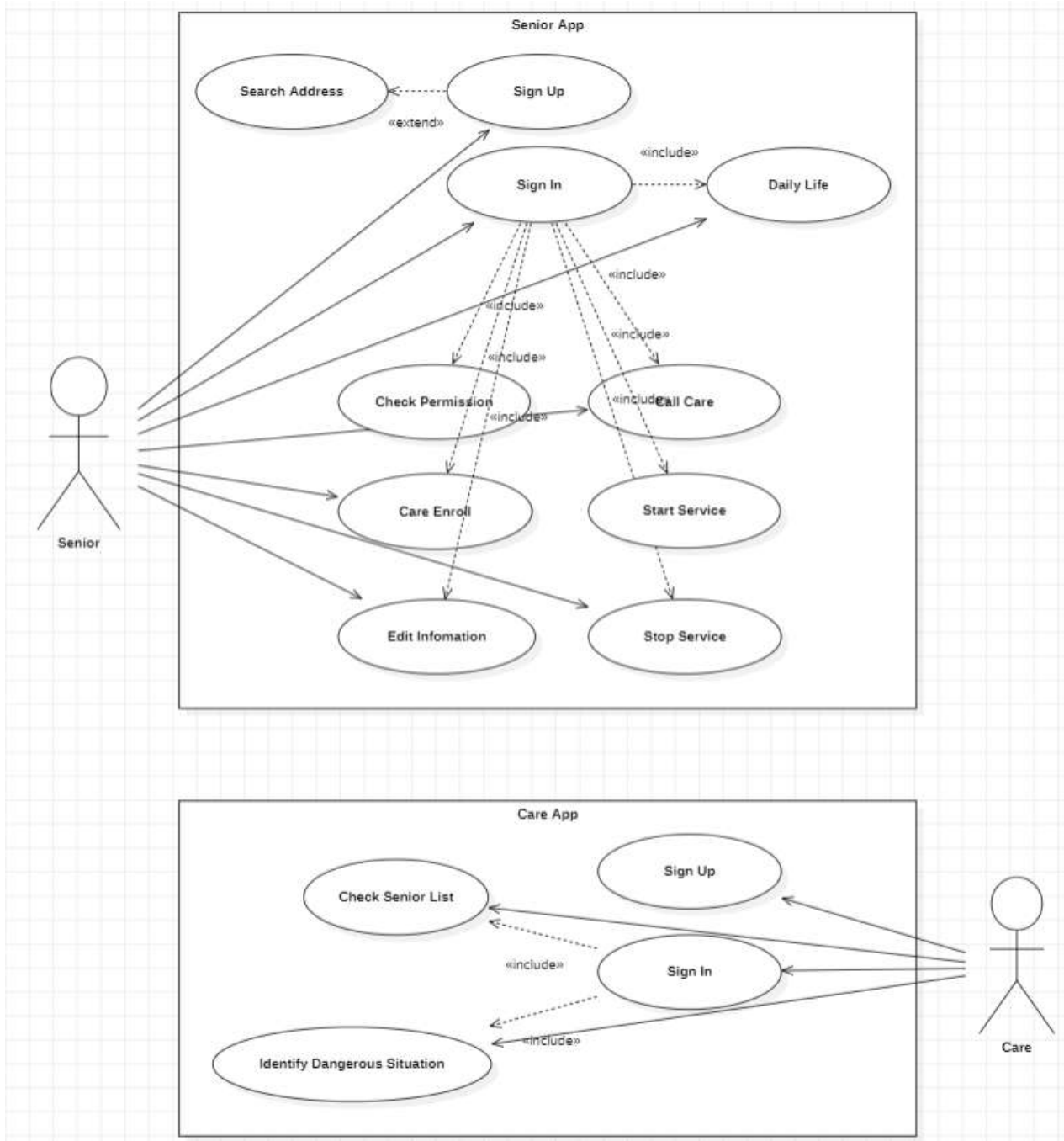
2.6 하드웨어 설계

(해당사항 없음)

2.7 소프트웨어 설계

가. 유스케이스

◇ 유스케이스 다이어그램



◇ 액터 목록

액터명	구분	설명
Senior	사용자	AI 스피커로 Senior 앱을 사용하는 사용자
Care	사용자	스마트폰으로 Care 앱을 사용하는 사용자

◇ 유즈케이스 설명

이름	CheckPermission	관련 액터	Senior
설명	사용자가 권한 설정을 하는 유즈케이스		
사건흐름	기본 흐름	1. 권한 설정을 요구하는 박스가 생성된다. 2. 권한 설정 허용 버튼을 누른다.	
	대안 흐름	1.a. 권한 설정을 이미 모두 했다면 박스는 생성되지 않고 바로 유즈케이스를 종료한다.	
	예외 흐름	2.a. 허용을 누르지 않고 취소 버튼이나 뒤로 가기를 한다면 앱은 종료된다.	
조건	사전 조건	SignIn	
	사후 조건	없음	
참고사항			

이름	SignUp	관련 액터	Senior / Care
설명	사용자가 회원가입을 하는 유즈케이스		
사건 흐름	기본 흐름	1. 사용자 정보를 입력한다. 2. 하단의 “가입하기” 버튼을 누른다.	
	대안 흐름	1.a. Senior인 경우, 주소를 포함한 모든 정보를 필수로 입력하여 가입한다. 1.b. Care인 경우, 주소를 제외한 모든 정보를 필수로 입력하여 가입한다.	
	예외 흐름	2.a. 이미 존재하는 ID, 비밀번호일 경우 오류 메시지를 출력한다. 2.b. 사용자 정보 형식이 올바르지 않은 경우 오류 메시지를 출력한다.	
조건	사전 조건	없음	
	사후 조건	로그인 액티비티로 이동한다.	
참고사항			

이름	SearchAddress	관련 액터	Senior
설명	Senior 사용자가 주소를 검색하는 유즈케이스		
사건흐름	기본 흐름	1. 주소를 검색한다. 2. 주소를 선택한다.	
	대안 흐름	없음	
	예외 흐름	없음	
조건	사전 조건	없음	
	사후 조건	없음	
참고사항			

이름	EditInfomation	관련 액터	Senior
설명	Senior 사용자가 자신의 정보를 수정하는 유즈케이스		
사건흐름	기본 흐름	1. 수정할 정보를 입력한다. 2. 하단의 수정 버튼을 눌러 정보를 수정한다.	
	대안 흐름	없음	
	예외 흐름	2.a. 수정한 사용자 정보의 형식이 올바르지 않은 경우, 오류 메시지를 출력한다.	
조건	사전 조건	SignIn	
	사후 조건	없음	
참고사항			

이름	SignIn	관련 액터	Senior / Care
설명	사용자가 로그인을 하는 유즈케이스		
사건흐름	기본 흐름	1. 사용자가 아이디와 비밀번호를 입력한다. 2. 로그인 버튼을 눌러 로그인한다.	
	대안 흐름	2.a. Senior 앱에서 보호자 정보를 등록했다면 메인 액티비티로 이동한다. 2.b. Senior 앱에서 보호자 정보를 등록하지 않았다면 보호자 등록 창으로 이동한다. 2.c. Care앱에서 로그인했을 경우 메인 액티비티로 이동한다.	
	예외 흐름	2.a. 아이디나 비밀번호가 올바르지 않은 경우 오류 메시지를 출력한다.	
조건	사전 조건	없음	
	사후 조건	없음	
참고사항			

이름	CareEnroll	관련 액터	Senior
설명	Senior 사용자가 보호자를 등록하는 유즈케이스		
사건흐름	기본 흐름	1. 사용자가 보호자 아이디를 입력한다. 2. 등록 버튼을 눌러 정보를 등록한다.	
	대안 흐름	없음	
	예외 흐름	2.a. 존재하지 않는 보호자 아이디를 입력했을 경우, 오류 메시지를 출력한다.	
조건	사전 조건	SignIn	
	사후 조건	없음	
참고사항			

이름	StopService	관련 액터	Senior
설명	Senior 사용자가 서비스를 중지하는 유즈케이스		
사건흐름	기본 흐름	1. 사용자가 “서비스 중지” 버튼을 누른다. 2. 서비스가 중지된다.	
	대안 흐름	없음	
	예외 흐름	없음	
조건	사전 조건	SignIn, 서비스가 동작 중인 경우	
	사후 조건	없음	
참고사항			

이름	StartService	관련 액터	Senior
설명	Senior 사용자가 서비스를 동작시키는 유즈케이스		
사건흐름	기본 흐름	1. 사용자가 “서비스 동작” 버튼을 누른다. 2. 서비스가 동작한다.	
	대안 흐름	없음	
	예외 흐름	없음	
조건	사전 조건	SignIn, 서비스가 중단된 경우	
	사후 조건	없음	
참고사항			

이름	CallCare	관련 액터	Senior
설명	Senior 사용자가 보호자를 호출하는 유즈케이스		
사건 흐름	기본 흐름	1. 사용자가 보호자 호출 버튼을 누른다. 2. 보호자 앱으로 알림을 보낸다.	
	대안 흐름	없음	
	예외 흐름	2.a 인터넷에 연결되지 않은 경우에는 메시지를 보낼 수 없기 때문에 오류 메시지를 출력한다.	
조건	사전 조건	SignIn, 인터넷이 연결되어있는 경우	
	사후 조건	없음	
참고사항			

이름	DailyLife	관련 액터	Senior
설명	Senior 사용자가 일상생활을 하는 유즈케이스		
사건흐름	기본 흐름	1. 사용자가 단말기 앞에서 행동을 취한다. 2. 카메라 프리뷰 화면에 현재 행동이 출력된다.	
	대안 흐름	1.a. 낙상이 의심되면 알림 메시지를 보호자 측에 보낸다.	
	예외 흐름	없음	
조건	사전 조건	SignIn, 서비스가 동작중인 경우	
	사후 조건	없음	
참고사항			

이름	CheckSeniorList	관련 액터	Care
설명	Care 사용자가 관리하는 Senior를 보여주는 유즈케이스		
사건흐름	기본 흐름	1. 관리 중인 Senior의 이름, 전화번호와 주소를 띄운다.	
	대안 흐름	없음	
	예외 흐름	1.a. 관리하는 Senior가 없는 경우 디스플레이에 “없음”이라고 출력한다.	
조건	사전 조건	SignIn	
	사후 조건	없음	
참고사항	Care 앱의 메인 액티비티		

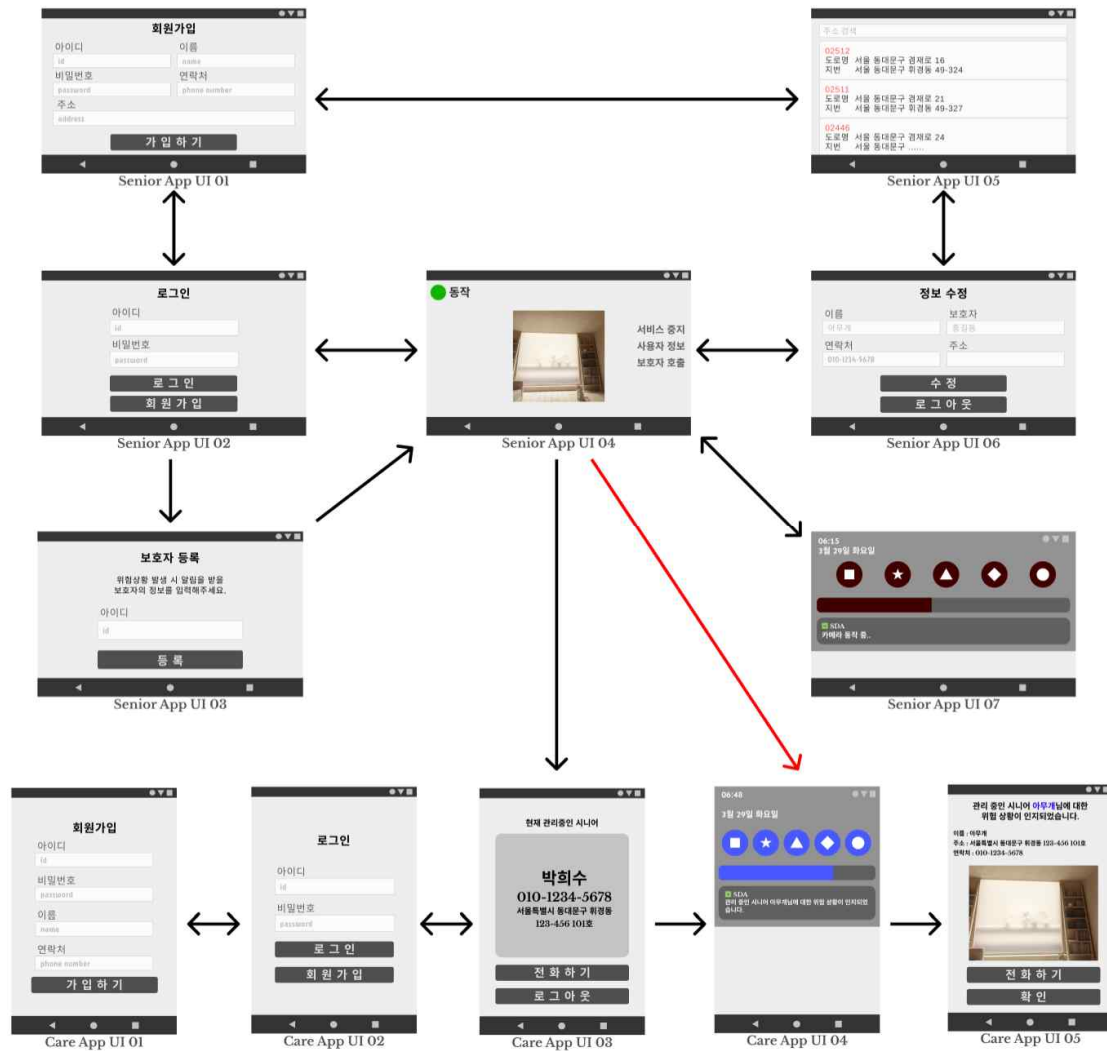
이름	IdentifyDangerousSituation	관련 액터	Care
설명	Care 사용자가 Senior의 위험상황을 인지하게 되는 유즈케이스		
사건흐름	기본 흐름	1. Care 사용자가 알림 메시지를 누른다. 2. Senior에 대한 정보와 위험 상황 당시의 이미지를 확인한다.	
	대안 흐름	없음	
	예외 흐름	없음	
조건	사전 조건	SignIn, Senior가 위험상황 알림을 보낸 경우	
	사후 조건	없음	
참고사항			

나. UI

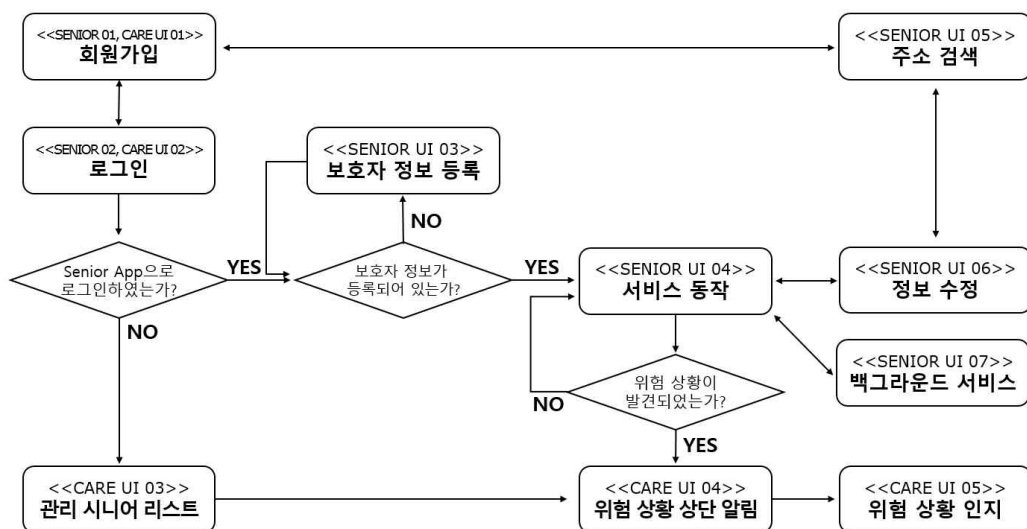
◇ 화면 목록

UI	화면	설명	구성 요소
SENIOR UI 01 / CARE UI 01	회원가입	회원가입 화면	입력창 : ID, Password, 이름, 연락처, 주소(시니어앱) 버튼 : 가입하기
SENIOR UI 02 / CARE UI 02	로그인	로그인 화면	입력창 : ID, Password 버튼 : 로그인, 회원가입
SENIOR UI 03	보호자 정보 등록	위험 상황 발생 시 알림을 받을 보호자 정보 등록	입력창 : 보호자 아이디 버튼 : 등록
SENIOR UI 04	서비스 동작	카메라가 작동 중이고 AI가 분석을 하고 있는 상태 카메라 프리뷰 제공	버튼 : 정보 수정, 서비스 중지, 보호자 호출 텍스트 : 서비스 동작 여부 뷰 : 카메라 프리뷰
SENIOR UI 05	주소 입력	주소 검색 API를 사용하여 도로명, 지번을 검색하여 입력	Javascript Web View 로드
SENIOR UI 06	정보 수정	이름, 보호자, 연락처, 주소 등의 사용자 정보를 수정	입력창 : 이름, 보호자, 연락처, 주소 버튼 : 수정, 로그아웃
SENIOR UI 07	백그라운드 서비스	앱이 꺼지거나 내려가도 백그라운드로 동작하는 상태 (상단 알림 제공)	상단 알림 메시지
CARE UI 03	관리 시니어 리스트	보호자가 현재 관리 중인 시니어 대상자 리스트를 확인할 수 있는 화면	리스트 : 관리 중인 시니어 버튼 : 전화하기, 로그아웃
CARE UI 04	위험 상황 상단 알림	관리 중인 시니어에게 위험 상황이 인지되었을 때 상단 알림으로 발생	상단 알림 메시지
CARE UI 05	위험 상황 인지	낙상이 발생한 시니어에 대한 정보와 당시 이미지 제공	텍스트 : 시니어의 정보 이미지뷰 : 당시 이미지 버튼 : 전화하기, 확인

◇ UI FLOW



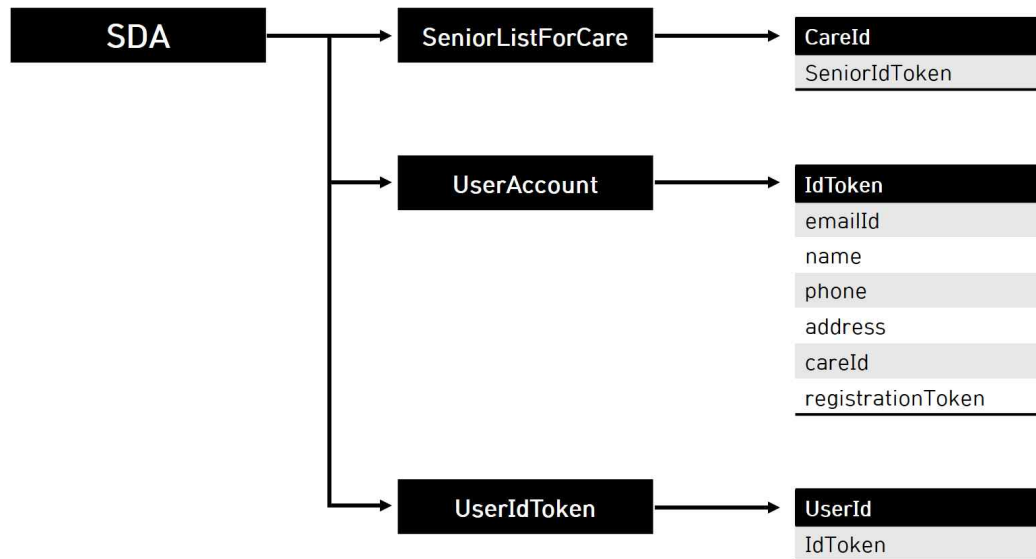
< FIGMA >



< UI FLOW >

다. Firebase Realtime Database

◇ DB 구성도



< DB 구성도 >

SDA	<pre> "SeniorListForCare":{ "CareId":"SeniorIdToken", ... } "UserAccount":{ "IdToken":{ "emailId":"email Id", "name":"user name", "phone":"phone number", "address":"address info", "careId":"care id for senior account", "registrationToken":"registration token value" }, ... } "UserIdToken":{ "userId":"idToken" ... } </pre>
-----	--

3. 결과 및 평가

3.1 완료작품 소개

가. 프로토타입 사진 (단말기 앱)

◇ 회원가입 (SENIOR UI 01)

회원가입

아이디 이름

id name

비밀번호 연락처

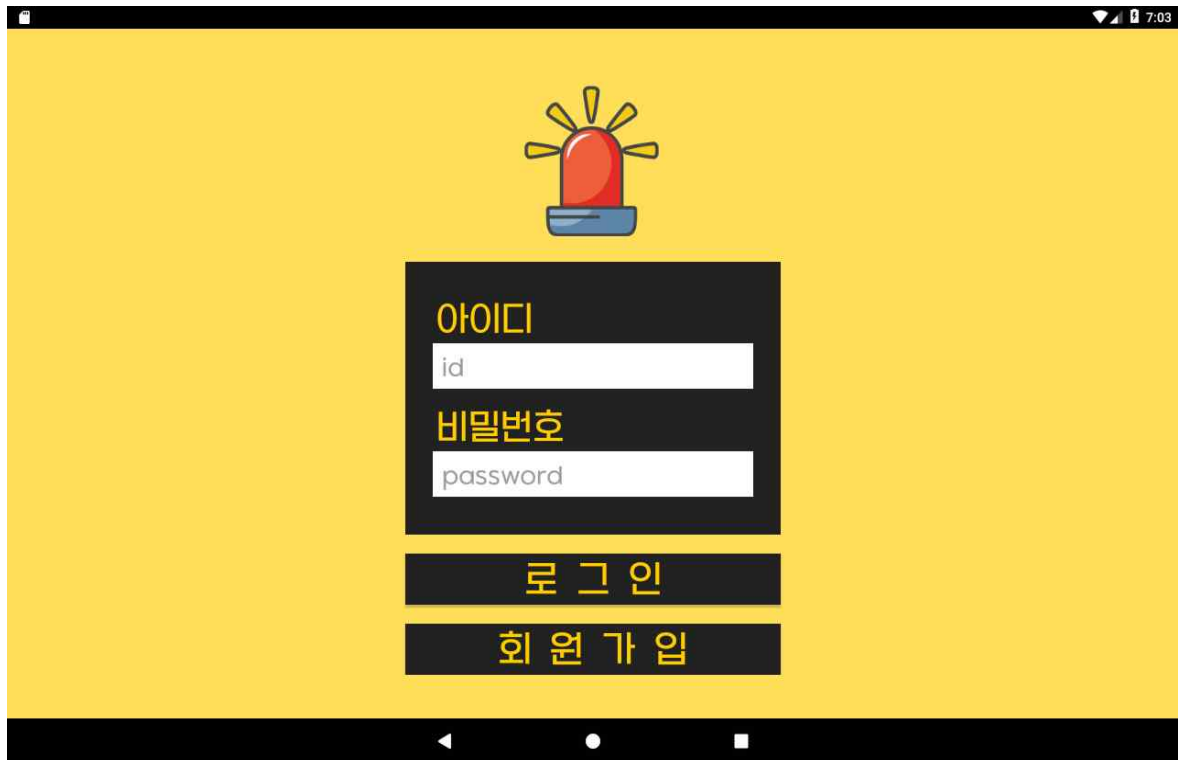
password phone number

주소

address

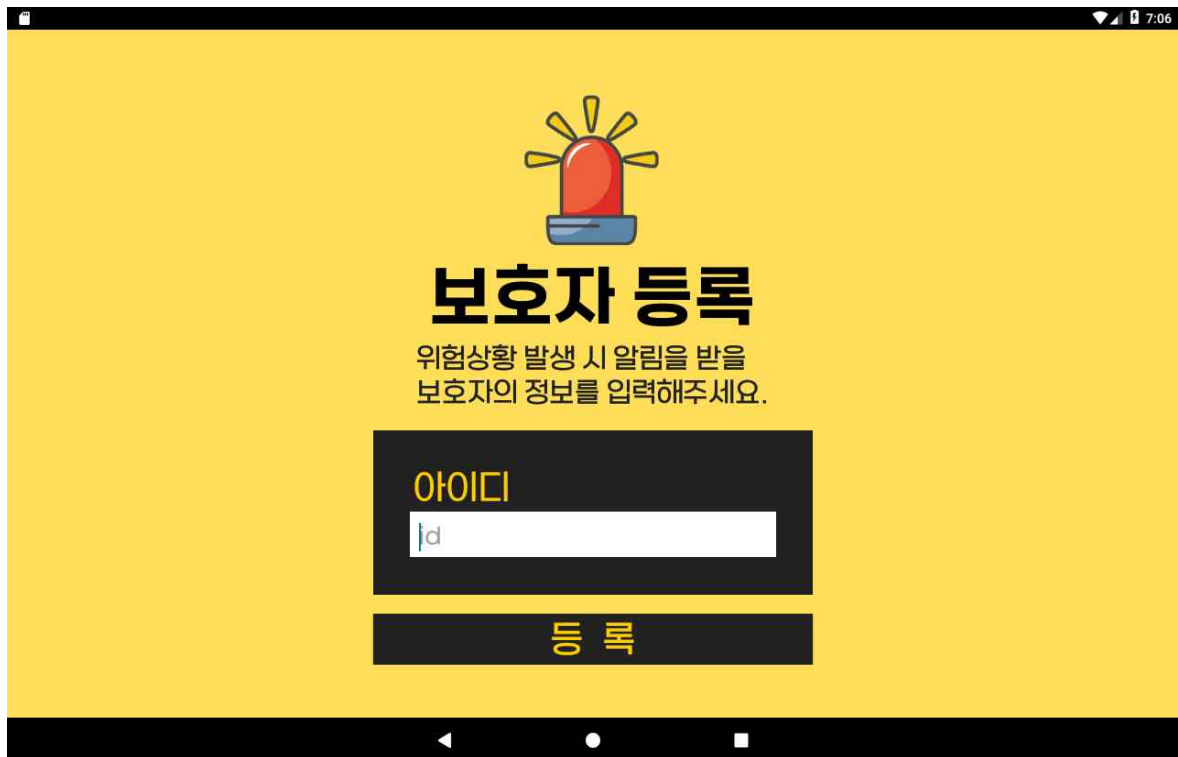
가입하기

◇ 로그인 (SENIOR UI 02)



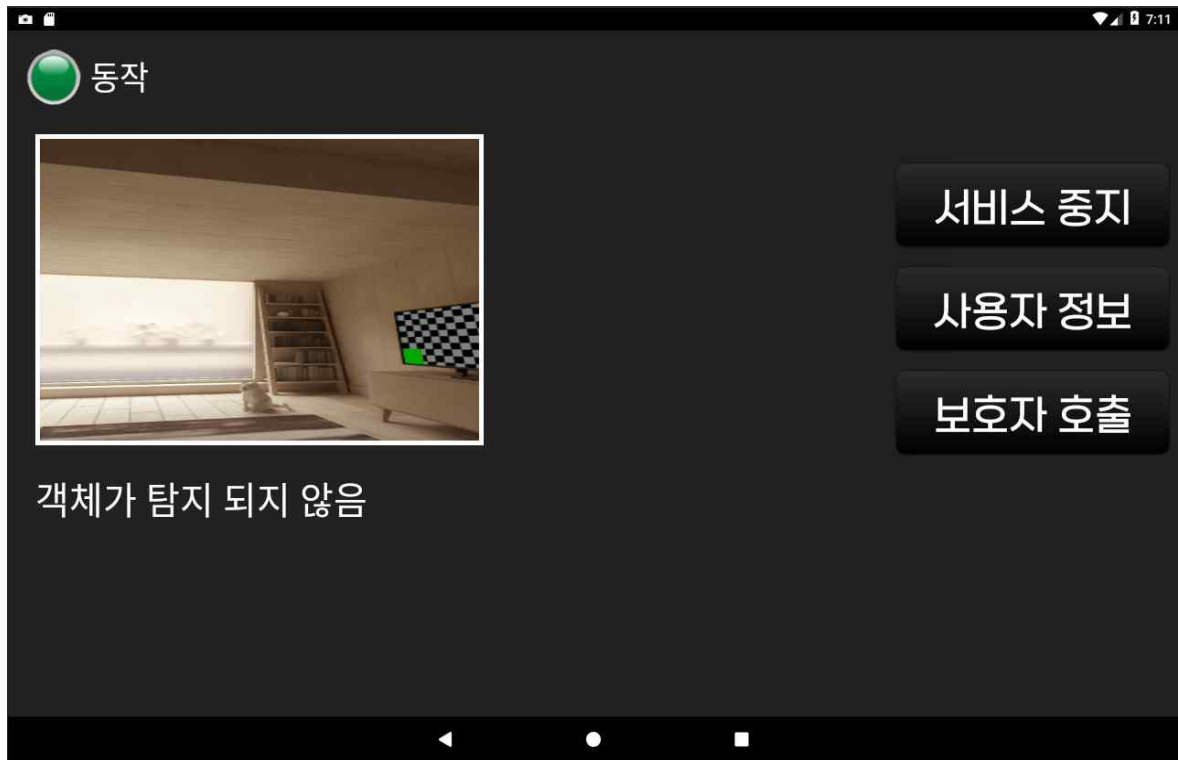
A mobile app login screen with a yellow background. At the top center is a red alarm bell icon with yellow sound waves. Below it is a dark gray rectangular box containing two input fields. The first field is labeled '아이디' (ID) in yellow and contains the text 'id'. The second field is labeled '비밀번호' (Password) in yellow and contains the text 'password'. Below these fields are two dark gray buttons with yellow text: '로그인' (Login) and '회원가입' (Sign Up). The screen is framed by a black status bar at the top showing signal, battery, and time (7:03), and a black navigation bar at the bottom with three white icons.

◇ 보호자 정보 등록 (SENIOR UI 03)

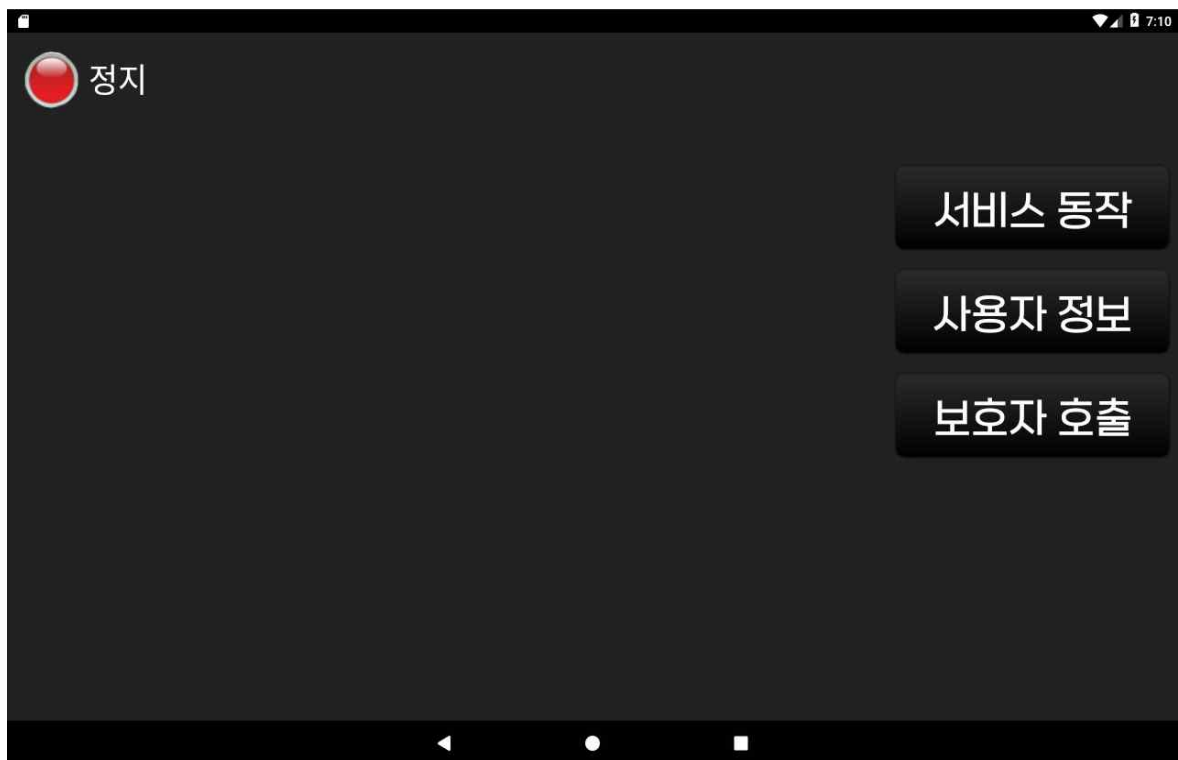


A mobile app screen for parent information registration with a yellow background. At the top center is a red alarm bell icon with yellow sound waves. Below it is the title '보호자 등록' (Parent Registration) in large black font. Under the title is a subtitle in black: '위험상황 발생 시 알림을 받을 보호자의 정보를 입력해주세요.' (Please enter the information of the guardian who will receive notifications in case of an emergency). Below this is a dark gray rectangular box containing one input field labeled '아이디' (ID) in yellow, which contains the text 'id'. At the bottom of the box is a dark gray button with yellow text: '등록' (Register). The screen is framed by a black status bar at the top showing signal, battery, and time (7:06), and a black navigation bar at the bottom with three white icons.

◇ 서비스 동작 (SENIOR UI 04)



◇ 서비스 중지 (SENIOR UI 04-02)



◇ 주소 입력 (SENIOR UI 05)

예) 판교역로 235, 분당 주공, 삼성동 681

tip
아래와 같은 조합으로 검색을 하시면 더욱 정확한 결과가 검색됩니다.

도로명 + 건물번호
예) 판교역로 235, 제주 첨단로 242

지역명(동/리) + 번지
예) 삼성동 681, 제주 영평동 2181

지역명(동/리) + 건물명(아파트명)
예) 분당 주공, 연수동 주공3차

사서함명 + 번호
예) 분당우체국사서함 1~100

Powered by kakao 우편번호 서비스 안내

◇ 정보 수정 (SENIOR UI 06)

정보수정

이름
ALICE

연락처
01012345678

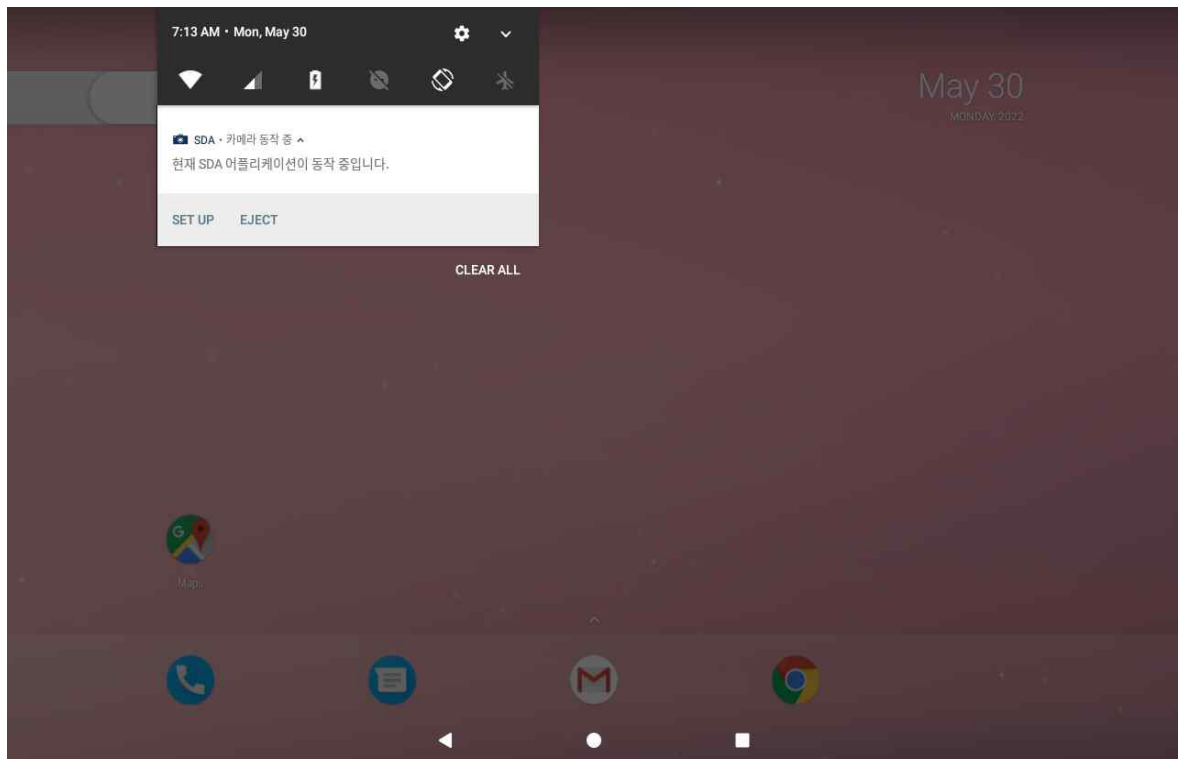
보호자
care02

주소
서울 종로구 대학로 138 (회화동)

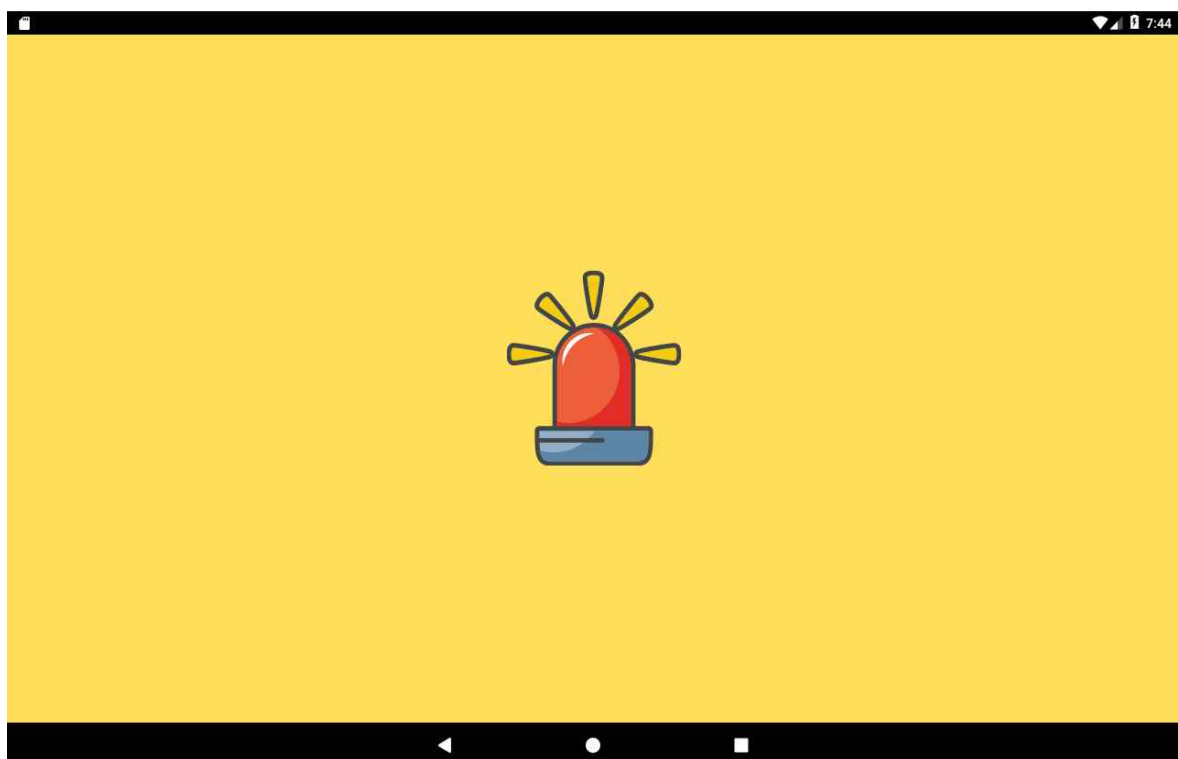
수정

로그아웃

◇ 백그라운드 서비스 (SENIOR UI 07)



◇ 스플래시 화면 (SENIOR UI SPLASH)



나. 프로토타입 사진 (보호자 앱)

◇ 회원가입 (CARE UI 01)

◇ 로그인 (CARE UI 02)

◇ 관리 시니어 정보 (CARE UI 03)

◇ 관리 시니어가 없을 때 (CARE UI 03-02)

◇ 위험 상황 상단 알림 (CARE UI 04)



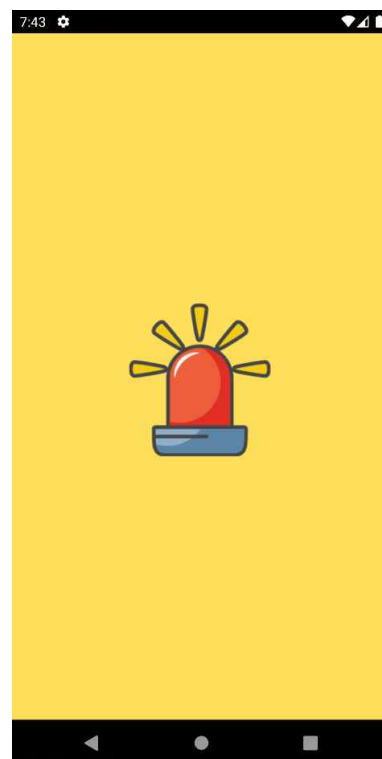
◇ 시니어가 호출 시 상단 알림 (CARE UI 04-02)



◇ 위험 상황 인지 (CARE UI 05)



◇ 스플래시 화면 (CARE UI SPLASH)



다. 낙상 감지 알고리즘

◇ 옆 방향으로 쓰러졌을 때 낙상 감지1



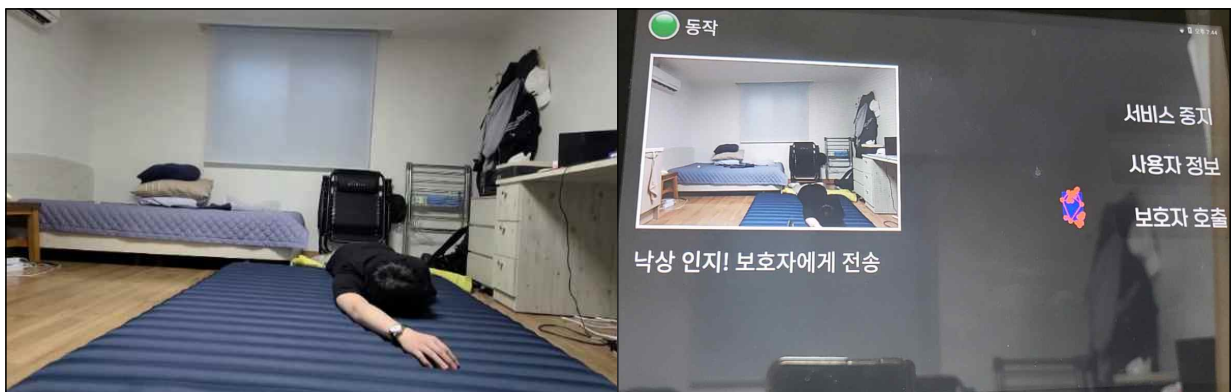
1차 판단, 2차 판단을 모두 충족하여 낙상으로 인지

◇ 옆 방향으로 쓰러졌을 때 낙상 감지2



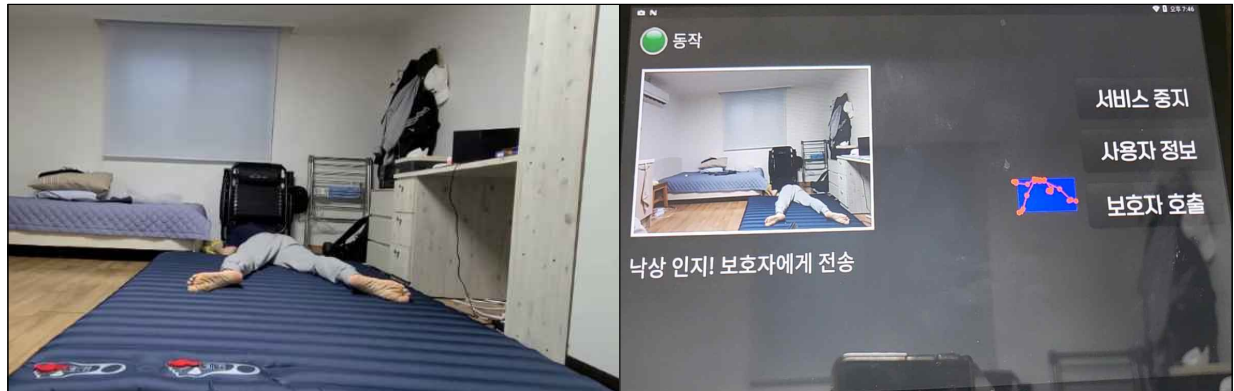
1차 판단, 2차 판단을 모두 충족하여 낙상으로 인지

◇ 카메라 쪽에 머리를 두고 쓰러졌을 때 낙상 감지



1차 판단, 2차 판단을 모두 충족하여 낙상으로 인지

◇ 카메라 쪽에 발을 두고 쓰러졌을 때 낙상 감지



1차 판단, 2차 판단을 모두 충족하여 낙상으로 인지

◇ 낙상 없이 일상생활1



낙상 인지를 하지 않음

◇ 낙상 없이 일상생활2



낙상 인지를 하지 않음

3.2 설치 (configuration)

가. SW/HW 구성 방법

◇ 단말기 앱 : SDA

- 기업에서 지원받은 단말기를 하드웨어로 사용한다.
- Android 7.1(API 수준 25)의 운영체제를 가지고 카메라를 사용할 수 있는 단말기를 사용한다.

◇ 보호자 앱 : SDA 보호자 앱

- Android 7.0(API 수준 24) 이상의 운영체제를 가진 스마트폰에서 사용할 수 있다.

나. 설치 방법

◇ 단말기 앱

- 단말기에 GMS가 없기 때문에 apk파일을 단말기에 받아서 설치해야 한다.

◇ 보호자 앱

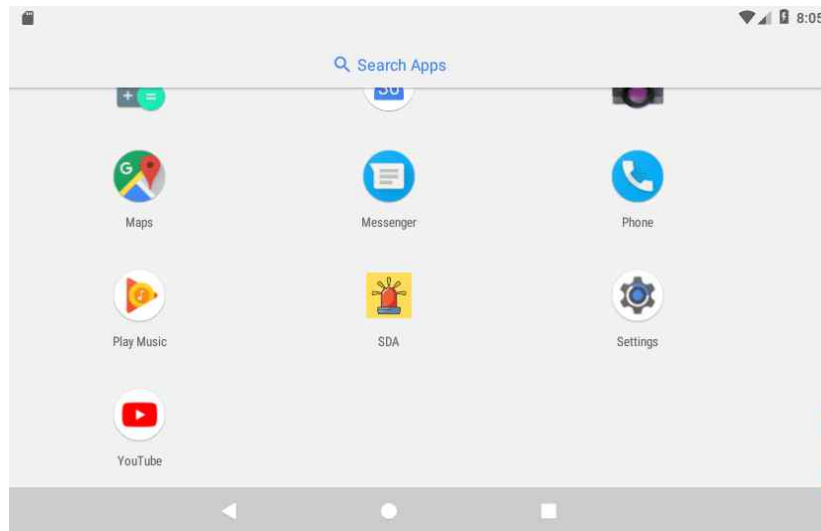
- 현재 플레이스토어에 올리지 않은 상태이기 때문에 apk파일을 기기에 받아서 설치해야 한다.
- 차후 플레이스토어에 올리게 되면 플레이스토어를 통해 설치할 수 있을 것이다.

3.3 실행 (run)

가. SW 실행 방법 (단계에 따른 화면 포함)

◇ 단말기 앱 : SDA

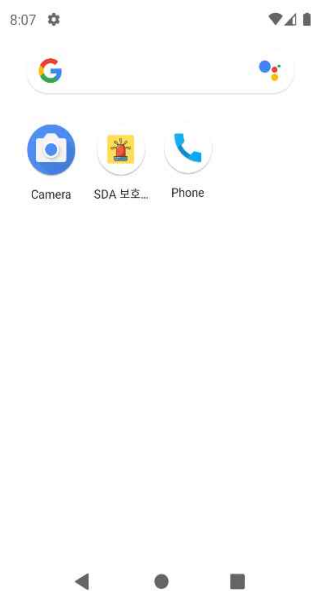
- 인터넷 연결을 확인한 후, 기기에 설치된 SDA 앱 아이콘을 눌러 실행한다. (수동 실행)



- 단말기가 재부팅되면, 자동으로 SDA 앱이 실행된다. (자동 실행)

◇ 보호자 앱 : SDA 보호자 앱

- 인터넷 연결을 확인한 후, 기기에 설치된 SDA 보호자 앱 아이콘을 눌러 실행한다.
- 단말기에서 알림을 보내면 상단 알림이 뜨고 알림을 누르면 앱이 실행된다.



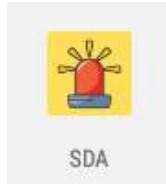
아이콘을 눌러 실행



상단 알림을 눌러 실행

3.4 성과물

◇ 단말기 앱 : SDA



◇ 보호자 앱 : SDA 보호자 앱



3.5 완료 작품의 평가

가. 평가 기준

평 가 항 목	평 가 방 법	적 용 기 준	개 발 목표치	비중 (%)
분류 정확도	테스트 셋 평가와 팀원들이 직접 상황을 재현해보며 평가	Accuracy, 100회 이상 평가	평균 90% 이상	40
모델 속도	입력값을 넣었을 때, 최종 결과까지의 출력 시간을 계산	System time, 200회 이상 측정	평균 10초 이하	40
응답 시간	위험 상황 데이터를 입력 후, 보호자 앱에 전송되는데 걸리는 시간 계산	System time, 50회 이상 측정	평균 5분 이하	20
			합계	100

나. 평가 결과

◇ 분류 정확도

방향	낙상 인지	횟수	정확도
전방	61	100	61%
후방	63	100	63%
좌측	78	100	78%
우측	80	100	80%

◇ 모델 속도

모델	실행 횟수	평균 속도(ms)
Pose Detection API	2,000	0.5105
3D Pose Baseline	2,000	4.706
ST-GCN	500	743.102

◇ 응답 시간

모델	실행 횟수	응답 시간(s)
보호자 호출 알림	50	2.068
위험 상황 알림	50	3.242

다. 평가 결과 분석

◇ 분류 정확도

낙상에 대한 상황이 아닌 일상 생활에서도 1차 알고리즘을 통과하는 경우가 있었다. 1차 알고리즘이 통과하는 원인을 보니 인간이 아닌 물건을 인간 객체로 인식하는 경우, 낙상이 아닌 상황이지만 1차 조건식을 통과하는 행동(ex. 다리를 벌리고 몸을 숙이는 행동) 등이 있었다. 하지만 이런 경우에는 1차는 통과하지만 쓰러져있는 상황이 아니기 때문에 2차 판단인 딥러닝 모델을 통과하지 못하여 최종적으로는 낙상으로 판단하지 않았다.

낙상에 대한 분류 정확도를 전,후,좌,우 방향으로 나누어 테스트를 진행하였다. 좌,우에 비해서 전,후의 정확도가 더 낮은 것을 확인할 수 있었다. 이는 1차적으로 스켈레톤 구조를 추출하는 Pose Detection API의 한계였다. Pose Detection API는 얼굴로 인간 객체를 찾고, 얼굴을 기반으로 나머지 관절 정보를 추출한다. 하지만 전방, 후방으로 낙상을 하는 경우에는 카메라에 얼굴이 잘 안보이는 경우가 많았고, 이 때문에 아예 관절 정보를 추출하지를 못하였다. 이러한 경우 때문에 전,후방의 정확도가 좌,우에 비해서 낮았다. 좌,우로 넘어지는 경우에는 전,후방에 비해 얼굴이 카메라에 나와 객체를 잘 탐지할 확률이 확실히 높아 낙상을 잘 인지하였지만, 관절 정보를 추출할 수 없을 정도로 얼굴이 보이지 않는 경우가 있어 낙상으로 인지되지 않을 때가 있었다.

딥러닝 모형 기반인 2차 판단의 성능은 train_set과 test_set에서 정확도는 0.9971, 0.9961로 높은 성능을 보였다. 하지만, 이는 잘 구성된 학습 데이터 상에서의 성능이었고, 실제 AI 단말기에서 촬영된 영상을 가지고 여러가지 행동을 취해보며 성능을 측정했다. 기존의 학습 데이터에서 존재했던 47개의 행동 종류 외의 다른 행동을 취했을 경우 모델이 해당 행동을 잘 처리하지 못하는 문제가 존재했다. 학습 데이터가 수집될 때 사용되었던 카메라의 성능, 각도가 현재 AI 단말기에서 사용되는 카메라와 다르다는 점도 성능 하락의 문제로 생각되었다. 실생활에서 예측할 수 없는 수많은 행동들을 잘 분류할 수 있도록, 여러가지 행동들이 있는 대규모 영상 데이터셋을 통해 추가적인 학습이 진행되어야 할 것이라고 판단되고, 모델의 최적화를 더 진행하여 강건한 시스템을 만들어야 된다고 판단된다.

학습 데이터 환경에서의 모델 평가 결과

	Predict 0	Predict 1
Target 0	18804	26
Target 1	35	2507

Train 예측 정확도 : 0.9971

	Predict 0	Predict 1
Target 0	4699	11
Target 1	10	623

Test 예측 정확도 : 0.9961

* Target : 실제 영상에 라벨링된 값으로, 0은 쓰러지기아닌 행동, 1은 쓰러지신 행동을 의미

* Predict : 입력에 대한 모델의 추론 값으로, 0은 쓰러지기아닌 행동, 1은 쓰러지신 행동을 의미

3.6 향후 평가

가. 어려웠던 내용들

- 1) FCM을 이용해 양방향으로 통신 보내기
- 2) 늑기와 낙상을 정확히 판별하기
- 3) 단말기 앱 내부에서 모든 모델이 처리되는 데에 긴 시간 소요
- 4) 모델을 돌리는 동안 사진 데이터를 저장하고 있을 메모리 공간 부족
- 5) 학습데이터 전처리 및 학습에 긴 시간 소요
- 6) 앞, 뒤로 쓰러지는 것에 대한 처리

나. 차후 구현할 내용

- 1) 학습 데이터를 추가적으로 구성하여 현재 어르신들의 47종류의 일상행동 이외의 행동들도 모델에 학습시키고, 이진 분류 모델을 다중 분류 모델로 구현한다.
- 2) 시니어 여러 명과 보호자 한 명을 연결할 수 있도록 구현한다.
- 3) 위험 상황 전에 나타날 수 있는 전조 증상에 대한 행동 인식을 추가하여 모델을 학습한다.

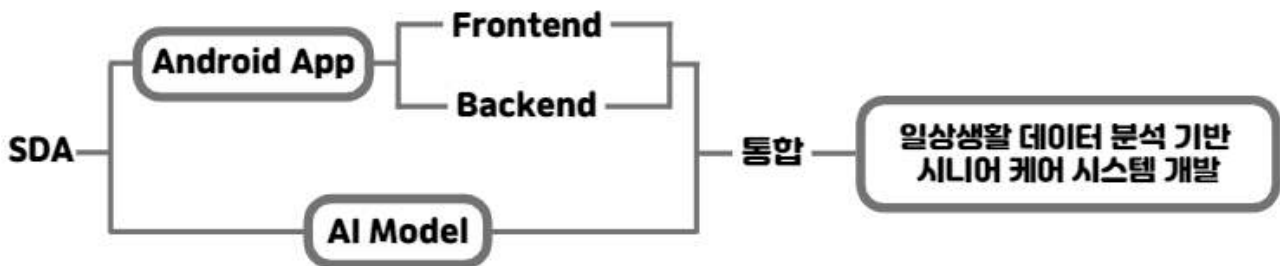
4. 개발 사업비 정산

4.1 구성원 및 추진체계

◇ 구성원

이름	책임	세부 내용
박희수 (팀장)	프로젝트 관리자 안드로이드 앱 개발 보조	1. 전체 프로젝트 관리 및 일정 조율 2. 최종 검토 및 승인 3. 단말기와 스마트폰 사이의 통신 구현 4. 파이어베이스 기능 연결
김현창	안드로이드 앱 개발	1. Android Application 작성 2. 시니어앱&보호자 앱 총괄 3. 카메라 제어 및 딥러닝 모델 이식 4. UI 디자인
정윤조	안드로이드 앱 개발	1. Android Application 작성 2. 단말기와 스마트폰 앱 통신 구현 3. 파이어베이스 기능 연결
최성수	딥러닝 모델 개발	1. 데이터 전처리 2. 모델 구축 및 학습 3. 모델 평가 4. Android에 이식가능한 형태로 모델 배포

◇ 추진체계



4.2 자재소요서

(해당사항 없음)

4.3 개발사업비 내역서

(해당사항 없음)

부 록

가. 참고문헌 및 참고사이트

- [1] 김성현, 김용욱, 권대규, 김남균(2006), “낙상 방향에 따른 신체 관절의 동적 특성 분석”, p447-p448
- [2] 김선기, 안종수, 김원호(2016), “영상처리 기반 낙상 감지 알고리즘의 구현”, p56-p59
- [3] 김지민, 윤기범, 심정용, 박소영, 신연순(2020), “YOLOv3 알고리즘을 이용한 실시간 낙상 검출”
- [4] 강윤규, 강희용, 원달수(2021), “PoseNet과 GRU를 이용한 Skeleton Keypoints 기반 낙상 감지”, p127-p132
- [5] 임태규, 『텐서플로 라이트를 활용한 안드로이드 딥러닝』, 한빛미디어(2021)
- [6] 장재영 외 4인, 지능형 행동인식 기술을 이용한 실시간 동영상 감시 시스템 개발 (The Journal of The Institute of Internet, Broadcasting and Communication (IIBC) Vol. 19, No. 2, pp.161-168, Apr. 30, 2019. pISSN 2289-0238, eISSN 2289-0246)
- [7] Sijie Yan, Yuanjun Xiong, Dahua Lin. “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”
- [8] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, Matthias Grundmann, Google Research. “BlazePose: On-device Real-time Body Pose tracking”
- [9] Daniele Grattarola, Cesare Alippi. “Graph Neural Networks in TensorFlow and Keras with Spektral”
- [10] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. “A simple yet effective baseline for 3d human pose estimation”

◇ Foreground Service

- <https://developer.android.com/guide/components/foreground-services>

◇ Pose Detection API

- <https://developers.google.com/ml-kit/vision/pose-detection>

◇ Firebase

- <https://firebase.google.com/docs/auth>
- <https://firebase.google.com/docs/storage>
- <https://firebase.google.com/docs/database>
- <https://firebase.google.com/docs/hosting>
- <https://firebase.google.com/docs/cloud-messaging>

◇ Tensorflow Lite

- <https://www.tensorflow.org/lite>

◇ Article

- 노년층 증가로 '시니어 맞춤형 서비스' 출시하는 기업들
http://it.chosun.com/site/data/html_dir/2020/11/07/2020110701161.html
- '초고령사회' 진입 초읽기...이통3사, '시니어 케어' 사업 확대에 적극 나서
<https://www.greened.kr/news/articleView.html?idxno=289627>
- 홀로 죽음 맞는 '고독사' 3배 늘었다...43%는 '65세 이상'
<https://www.nocutnews.co.kr/news/5698160>
- 통화중 ATM 인출 등 이상행동...신한은행, AI로 보이스피싱 예방
<https://news.mt.co.kr/mtview.php?no=2022030710540742302>
- SKT, AI 기반 지능형 영상분석 솔루션 내년 상반기 출시...자체 AI칩으로 기술 격차 자신
<http://www.aitimes.com/news/articleView.html?idxno=141159>

◇ Data Resource

- 한국전자통신연구원(ETRI) : <https://ai4robot.github.io/etri-activity3d-livinglab/>
- Jinhyeok Jang, Dohyung Kim, Cheonshu Park, Minsu Jang, Jaeyeon Lee, Jaehong Kim, "ETRI-Activity3D: A Large-Scale RGB-D Dataset for Robots to Recognize Daily Activities of the Elderly", International Conference on Intelligent Robots and Systems (IROS) 2020, pp.10990-10997

나. 관련특허

- 엘에스산전 주식회사, 독거노인 케어 시스템, 10-2013-0053844, 2013
- 한남대학교 산학협력단, 리모콘을 이용한 위험 관리 시스템, 10-2016-0018784, 2016

다. 소프트웨어 프로그램 소스

[Github] <https://github.com/h-spear/Senior-Danger-Analysis>

[Github] <https://github.com/h-spear/Senior-Danger-Analysis-Care-App>

[Github] https://github.com/ttjh1234/CSproject_SDA