



부산대학교
PUSAN NATIONAL UNIVERSITY

2025 전기 졸업과제 착수보고서

청각 제약 상황을 위한 실내 소리 인식 및 상황 감지 시스템



부산대학교 정보컴퓨터공학부

202055546 박지용

202255666 이민경

202255669 이진솔

1. 연구 주제.....	3
2. 과제 배경 및 목표.....	3
2.1 과제 배경	3
2.2 개발 목표	3
3. 문제 정의 및 요구 사항 분석.....	4
3.1 문제 정의	4
3.2 요구 사항 분석.....	4
3.2.1 사용자 인식 및 추적	4
3.2.2 실내 지도 생성.....	4
3.2.3 소리 인식 및 분류.....	5
3.2.4 알림 시스템	5
3.2.5 소리 저장 및 관리.....	5
4. 시스템 구성.....	6
4.1 시스템 구성도	6
4.1.1 시스템 아키텍처.....	6
4.1.2 데이터베이스 ERD 모델	6
4.2 시스템 개발환경 및 사용 기술	7
4.2.1 개발 언어.....	7
4.2.2 개발 도구.....	7
4.2.3 통신 환경.....	8
4.2.4 실행 환경 및 요구사항.....	8
5. 현실적 제약 상황 및 대책.....	9
5.1 데이터 클래스 혼동문제.....	9
5.2 무선 통신 끊김 현상.....	10
5.3 SLAM 오류 가능성.....	10
6. 개발 일정 및 역할 분담.....	11
6.1 개발 일정	11
6.2 역할 분담	12

1. 연구 주제

청각 제약 상황을 위한 실내 소리 인식 및 상황 감지 시스템

2. 과제 배경 및 목표

2.1 과제 배경

오늘날 일상생활에서는 초인종, 경보음, 안내 방송 등 다양한 소리를 통해 정보가 전달되며, 이는 비장애인에게는 자연스럽게 직관적인 의사소통 수단으로 작용한다. 그러나 청각장애인이나 노약자와 같은 청각적 정보 인지가 어려운 계층은 이러한 소리 정보를 실시간으로 인식하고 반응하는 데 어려움이 있어, 위급 상황에서의 빠른 대처는 물론 방문자 응대나 방송 안내 인지에도 제약을 받는다. 특히 실내 환경에서는 소리의 방향이나 발생 위치를 정확히 파악하기 어렵기 때문에, 단순한 소리 감지만으로는 전체 상황을 이해하기 어려운 구조적 한계가 존재한다.

본 프로젝트는 해당 문제를 해결하기 위해 소리를 인식하고, 종류와 위치를 분석하며, 이를 시각적 정보로 변환하여 안내하는 시스템을 제안한다.

2.2 개발 목표

본 프로젝트는 사용자의 사진을 기반으로 사용자를 인식하고 추적하며, 처음에는 로봇에 달린 카메라를 이용해 실내 지도를 생성한다. 실내 환경에서 발생하는 주요 생활 소음과 경보를 카메라 배열을 통해 방향을 인식하고 이를 3단계 (위험, 주의, 도움)으로 분류한다. 분류된 소리는 모바일 앱을 통해 사용자에게 실시간으로 알림을 제공하고, 실내 지도에 소리 발생 위치를 시각적으로 표시한다. 로봇에서도 LED 색상을 통해 현재 어떤 소리가 났는지 사용자에게 바로 전달한다. 또한 소리 데이터를 저장하고 관리할 수 있는 아카이빙 기능을 통해 사용자가 지난 소리를 날짜, 시간별로 확인할 수 있도록 개발할 예정이다.

3. 문제 정의 및 요구 사항 분석

3.1 문제 정의

청각 장애인 및 노약자 등 청각 제약 상황에 있는 사용자는 화재 경보, 초인종, 경보음 등 중요한 소리 정보 뿐 아니라 물 떨어지는 소리나 전자제품의 알림음 같은 일상적인 생활소리까지도 인지하기 어렵다. 이로 인해 위험을 감지하지 못하거나 생활의 질이 저하되는 문제가 발생한다. 기존 소리를 시각적으로 보여주는 알림 시스템은 화재경보에 국한되어 만들어져 있으며 다양한 생활 소리에 대한 대응이 부족하다.

3.2 요구 사항 분석

3.2.1 사용자 인식 및 추적

- 로봇에 탑재된 카메라를 통해 사용자의 얼굴을 실시간으로 인식 및 추적할 수 있어야 한다.
- 앱 회원가입 시 얼굴 사진 등록이 필수이며, 등록된 이미지는 사용자 식별을 위한 레퍼런스로 사용된다.
- 초기 얼굴 인식이 실패할 경우, 로봇은 사용자의 얼굴을 포착하기 위한 주변 환경 스캔을 수행한다.
- 로봇은 사용자의 얼굴을 인식한 경우 Robomaster SDK의 사람 인식 기능을 통해 사용자를 추적한다.
- 사용자가 시야에서 이탈한 경우, 로봇은 마지막 인식 위치를 기준으로 재탐색하며 얼굴을 재식별한 뒤 추적을 재개한다.
- 일정 시간 이상 (ex 10분) 탐색해도 사용자를 찾을 수 없는 경우 로봇은 지정된 위치에서 대기한다.

3.2.2 실내 지도 생성

- 로봇은 Visual SLAM 기술을 이용해서 실내 공간을 스스로 파악하고, 지도를 만든다.
- 만들어진 지도는 앱에 표시되고, 사용자 위치, 로봇 위치, 마이크 위치, 마이크 기준 소리의 방향을 지도 위에 표시한다.

3.2.3 소리 인식 및 분류

- YAMNet이라는 사전 학습된 소리 분류 모델을 사용해서, 실내에서 나는 소리를 분류한다.
- 소리는 다음과 같이 세 가지 단계로 분류된다:
 - 위험: 화재 경보, 공습 경보 등 긴급 상황
 - 주의: 유리가 깨지는 소리, 낙상 소리 등 사용자 주의가 필요한 상황
 - 도움: 초인종, 전자레인지 알림음 등 일반적인 알림성 소리
- 마이크 어레이를 통해 소리의 방향(각도) 정보를 획득하고, 이를 라즈베리 파이에서 YAMNet을 이용해 소리를 분류한 뒤 TCP 소켓 통신을 통해 Xavier로 전송한다.
- 전달되는 데이터에는 소리 이벤트 종류와 해당 소리의 발생 방향(각도)이 포함된다.
- '주의' 및 '도움' 분류에 해당하는 소리는 방향 정보를 활용하여 지도 상에 위치를 시각화한다.

3.2.4 알림 시스템

- 분류된 소리 정보는 즉시 모바일 앱에 푸시 알림 형태로 전달된다.
푸시 알림에는 다음과 같은 정보가 포함되며, 글자와 이모티콘이 함께 전달된다.
 - 소리의 분류 등급: 위험 / 주의 / 도움
 - 소리의 종류: 예) 초인종 소리, 유리 깨짐 소리 등
- 사용자가 푸시 알림을 클릭하면, 앱 상세 페이지로 이동하여 추가 정보를 확인할 수 있다.
- 소리 분류 등급에 따라 로봇의 LED를 다음과 같이 점등한다.
 - 위험: 빨간불
 - 주의: 노란불
 - 도움: 초록불

3.2.5 소리 저장 및 관리

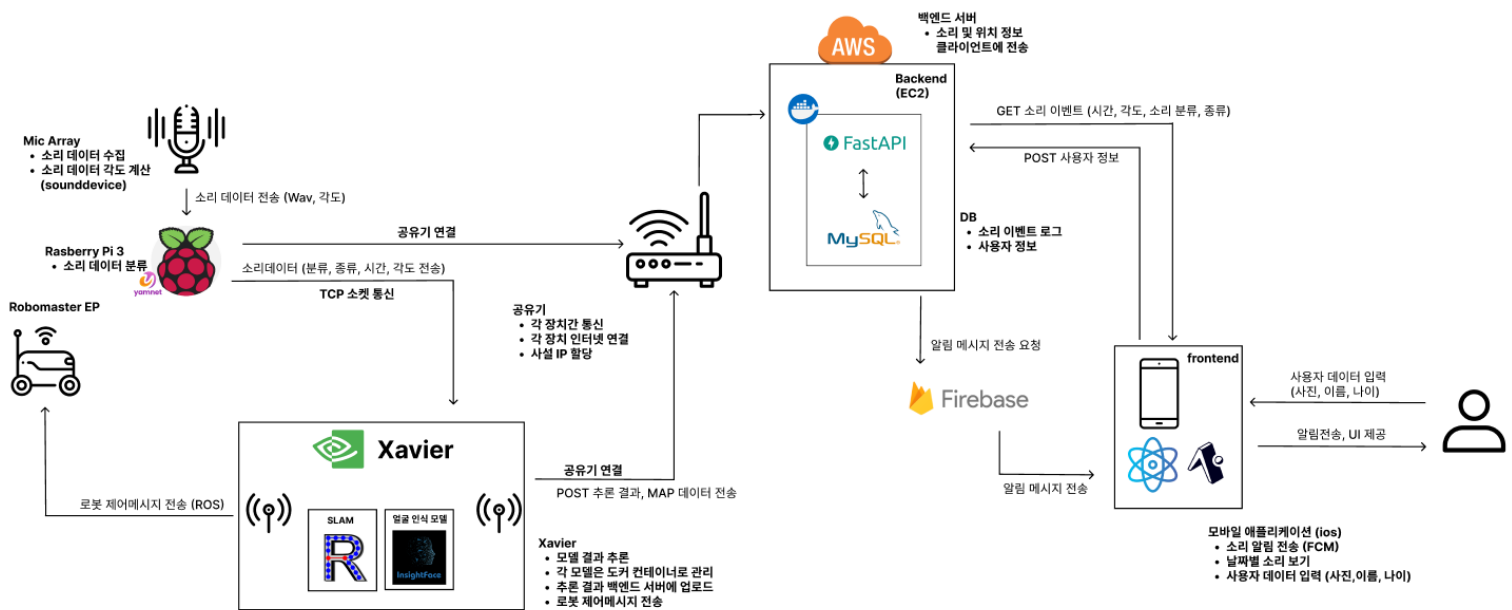
- 앱을 통해 발생했던 소리 이벤트를 확인할 수 있다.
- 앱에 진입하면 달력이 표시되며 각 날짜별 발생한 소리 이벤트가 해당 일자 칸 내 숫자로 표시된다.
- 해당 일자를 클릭하면 소리 리스트가 표시되며 최근 이벤트가 가장 상단으로 올라온다.
- 리스트에서는 소리 이벤트 종류에 따라 다른 이모티콘을 표시한다.
 - 위험 : 🚨
 - 주의 : ⚠️
 - 도움 : 💡
- 사용자가 리스트 중 하나를 클릭하면, 더 자세한 정보(시간, 위치, 종류 등)페이지로 이동한다. 페이지에는 다음과 같은 내용을 포함한다..

- 소리 이벤트 발생 위치 (실내 지도 그림 위 마이크 위치에서 방향 표시)
- 소리 이벤트 종류 이모티콘: 위험, 주의, 도움 내에서 한번 더 분류 (ex. 깨지는 소리 일 경우 🛑)

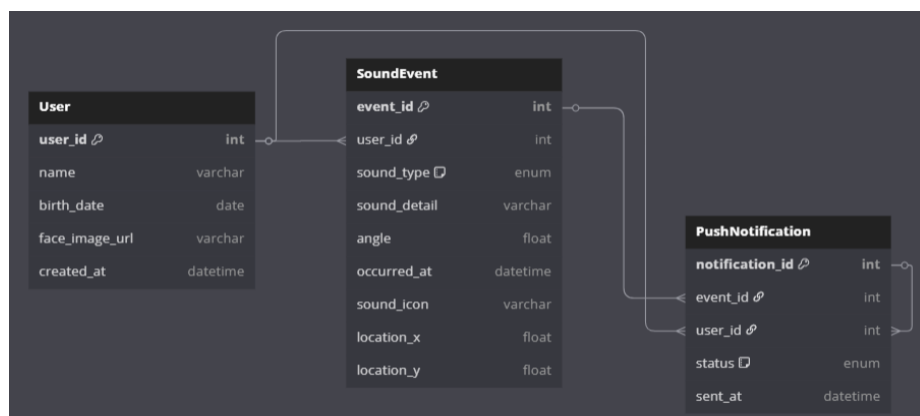
4. 시스템 구성

4.1 시스템 구성도

4.1.1 시스템 아키텍처



4.1.2 데이터베이스 ERD 모델



4.2 시스템 개발환경 및 사용 기술

4.2.1 개발 언어

- TypeScript : 모바일 앱, 알림 서비스
- Python: 백엔드, AI 모델, 로봇 제어
- SQL : 데이터베이스

4.2.2 개발 도구

모바일 앱 (클라이언트)	React Native, Expo
백엔드 서버 (API)	FastAPI
AI 모델 추론	YAMNet(소리분류), Insightface(얼굴인식)
실내 지도 생성	RTAB-Map(ROS 2 Humble)
딥러닝 프레임 워크	PyTorch 2.7.0, TensorFlow lite
로봇 제어	Robomaster SDK, ROS, RTAB-Map
데이터베이스	MySQL
인프라	Docker, AWS (EC2), Github Actions
알림	Firebase FCM
IDE / 편집기	VS Code(Python, Docker, CMake, C++)
협업 & 문서	Notion, google docs

4.2.3 통신 환경

통신 대상	방식	사용 도구	전달 데이터
모바일 앱 ↔ 백엔드 서버	RESTful API (HTTP), FCM (푸시 알림)	FastAPI(백엔드), Axois (프론트엔드), Firebase Cloud (알림)	소리 데이터 (분류, 정보, 위치, 시간), 사용자 정보 (이름, 생년월일, 사진)
Raspberry Pi (클라이언트) ↔ Xavier (서버)	TCP 소켓 통신	Python socket	소리 데이터 (분류, 정보, 시간)
Xavier ↔ 백엔드 서버	RESTful API (HTTP)	FastAPI(백엔드), Python Requests	소리 데이터 (분류, 정보, 시간, 발생 각도)
로봇 ↔ Xavier	Wifi(AP)	robomaster SDK	로봇 제어 명령

- TCP 소켓 통신 : Raspberry Pi의 전원이 들어오면 소켓 연결을 수립한다. Xavier는 Listen 상태로 연결을 기다린다.

4.2.4 실행 환경 및 요구사항

프로젝트는 다음과 같은 환경에서 구동된다.

1. 모바일 앱
 - 디바이스 : iPhone
 - 개발 플랫폼 : macOS + Xcode
2. 소리 분류
 - 하드웨어 : Raspberry Pi Model B+
 - 운영체제 : Raspberry Pi OS (Linux 기반)
 - 언어 및 런타임 : Python 3.9, tf lite_runtime
 - 오디오 장비 : Respeaker USB 4-Mic Array
 - 사용 라이브러리 : sounddevice (오디오 캡처용), numpy (오디오 전처리)
 - 사용 모델 : YAMNet(tflite) :
3. 얼굴 인식
 - 운영체제 : Ubuntu 22.04

- 언어 및 프레임 워크 : Python 3.10, ONNX Runtime 1.17.1, numpy 1.24.4, opencv-python 4.8.1.78
 - 모델 : buffalo_l (`w600k_r50.onnx`), Backbone: ResNet-50
 - 학습 데이터셋 : WebFace600k
4. SLAM
- 가상머신 : VirtualBox
 - 운영체제 : Ubuntu 22.04
 - 미들웨어 : ROS 2 Humble
 - 필수 패키지 : ros-dev-tools, ros-humble-rtabmap-ros, ros-humble-rtabmap-demos
5. 로봇
- 하드웨어 : DJI Robomaster EP
 - 제어 인터페이스 : Robomaster SDK
6. 백엔드 서버
- 언어 및 프레임워크: Python 3.10, FastAPI
 - 데이터베이스: MySQL 8.0

5. 현실적 제약 상황 및 대책

5.1 데이터 클래스 혼동문제

사전 학습된 YAMNet 모델은 '경보음(Alarm)'과 '벨소리(Bell/Ringtone)' 계열 간의 음향적 특징이 유사해 클래스 혼동이 자주 발생한다.

- 예시 1. 화재경보 사이렌 vs 타이머
- 예시 2. 스마트폰 긴급 재난 문자음 vs 휴대폰 벨소리

이처럼 클래스 간 경계가 명확하지 않은 음원에 대해서는, 기존 모델이 훈련된 데이터셋으로는 실제 환경에서의 소리 구분에 한계가 있다.

다음과 같은 방법으로 이러한 문제를 완화한다.

- 대책 1. 데이터 전처리 / 증강
화재경보, 비상벨, 초인종, 휴대폰 벨, 타이머 등 각 클래스별 50~100개 데이터를 수집하고, 거리, 음량이 다른 다양한 조건(실내/복도, 문 열림/닫힘 등)에서도 녹음해 데이터 다양성을 확보한다.
- 대책 2. Softmax 기반 출력 신뢰도 임계값 설정
모델의 출력 결과는 Softmax 확률값으로 표현되며, 이 중 일정 수준 이상의 신뢰도일때만 분류결과를 반영하도록 한다.

5.2 무선 통신 끊김 현상

로봇과 백엔드 간의 통신이 Wi-Fi 기반으로 이루어져 환경 고전에 따라 네트워크 끊김 현상이 발생할 수 있다. 이러한 경우 로봇이 제어 명령을 수신하지 못하거나 데이터를 송신하지 못해 정상적인 동작이 어려워지고 사용자가 해당 상황을 인지하지 못하는 문제가 발생할 수 있다.

다음과 같은 방법으로 이러한 문제를 완화한다.

- Xavier 대기 로직

일정 시간동안 로봇의 응답이 없을 경우, 사용자에게 알림을 전송하여 통신 장애 발생을 알린다.

- 로봇 복귀/정지 로직

통신이 끊겼다고 판단되면 로봇은 사전에 설정된 동작인 안전지대 복귀 또는 자동 정지를 수행한다. 그리고 LED 색상 변화를 통해 연결이 끊긴 상태인 것을 사용자가 시각적으로 확인할 수 있도록 한다.

5.3 SLAM 오류 가능성

로봇에 탑재된 카메라와 센서를 활용하여 실내 환경의 공간 정보를 수집하고, 이를 SLAM기술을 적용해 실내 지도를 생성한다. 그러나 실제 주거 공간의 구조적, 환경적 특성에 따라 SLAM 성능이 저하되거나 오류가 발생할 수 있는 변수들이 존재한다.

다음과 같은 방법으로 이러한 문제를 완화한다.

- 방법 1. SLAM 재수행

초기 SLAM 수행 이후에도 주기적으로(예: 일정 거리 이상 이동 시) SLAM을 재수행하여 지도를 점진적으로 보완한다.

- 방법 2. 사전 순회

시스템 초기화 시, 로봇이 공간 전체를 순회하는 사전 경로 탐색 모드를 제공해 지도 누락을 방지한다. 이 과정에서 주요 실내 공간(예: 거실, 주방, 복도 등)을 빠짐없이 스캔하여 초기지도의 완성도를 높이도록 한다.

6. 개발 일정 및 역할 분담

6.1 개발 일정

단계	작업	5월			6월					7월				8월				9월
		16	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	1
착수 및 환경 구축	개발환경/SDK 세팅																	
	하드웨어 테스트																	
	TCP 소켓 통신 구현																	
	얼굴등록 UI																	
핵심기능 개발	얼굴인식 POC 통합																	
	소리 데이터 수집 및 분류																	
	SLAM 지도생성																	
	백엔드 API 및 DB 설계																	
	EP SDK, ROS2 연동																	
	주행 알고리즘 개발																	
	모바일 앱 UI 설계 및 구현																	
	소리 + 각도 지도에 표시																	

E
N
D

통합 개발 및 기능확장	LED, 푸시알림 연동																	
테스트 및 결과정리	테스트																	
	버그수정																	
	보고서 작성																	

6.2 역할 분담

이름	역할
박지용	<ul style="list-style-type: none"> - 사용자 인식 및 실내 소리 분류 AI 개발 - 로봇 위치 추정 - SLAM 구현
이민경	<ul style="list-style-type: none"> - 주행 알고리즘 개발 - 앱 프론트 개발 - 시스템 인프라 구성
이진솔	<ul style="list-style-type: none"> - 백엔드 API 설계 - DB 설계 및 연동 - 데이터 수집 및 전처리