

# 컴퓨터공학부

## 졸업설계 최종결과보고서

과제명	딥러닝 기반 위험 소리 감지 앱
팀 장	김상우
팀 원	김성원, 손영진, 차주원, 김문웅

2023. 05. 30

지도교수 : 한연희

(인)

작 성 자 : 김상우

(인)

※지도교수 및 작성자 서명은 자필서명이 원칙(직인 사용 불가)

# 목 차

I . 서론 .....	2
1. 작품선정 배경 및 필요성 .....	2
2. 연구 목표 .....	2
3. 역할 분담 및 팀 소개 .....	3
4. 연구 일정 .....	3
II . 본론 .....	4
1. 연구 내용 .....	4
2. 문제점 및 해결방안 .....	9
3. 공학문제 수준 설명 .....	12
III . 결론 .....	13
1. 연구 결과 .....	13
2. 작품제작 소요재료 목록 .....	16
참고자료 .....	16

# I. 서론

## 1. 작품선정 배경 및 필요성

사람이 살아가면서 위험이 닥치는 상황을 아는 법은 오감을 사용하는 법이다. 그 중 일반적인 생활에 주로 사용되는 것은 시각과 청각이다.

그 중 시각에 제한이 있더라도 이를 보완하는 장치는 사회적, 체계적으로 구비되어 있다. 대부분의 거리에 있는 점자블록, 점자글, 횡단보도에서의 신호알림 등이 그 예다. 하지만 청각에 제한이 있는 경우 이를 보완할 장치는 쉽게 찾아보기 어려우며 그 예시도 들기 어렵다. 소리를 듣지 못하면 화재가 발생해 경보가 울리더라도 불이나 연기를 보기 전까지는 알 수 없어 탈출이 늦어 위험해 처한다. 또한 차가 아무리 뒤에서 경적을 울리더라도 알아차리지 못한다. 위험에 대하여 대처할 기회조차 얻지 못하는 상황이다. 현대에 와서는 미디어가 발달하면서 청각 장애인이 아니더라도 소리를 듣지 못 하는 경우가 있다. 날이 갈수록 이어셋이나 헤드셋 등으로 스스로의 귀를 막은 채 거리를 걷는 사람은 자주 보인다. 이런 사람들이 소리를 듣지 못해 사고가 나는 경우도 종종 뉴스로 나온다. 위의 일로 봤을 때 소리를 듣지 못하는 상황은 소리를 들을 때 피할 수 있는 사고를 피할 기회조차 앗아가는 것이다. 이를 보완하기 위해 위험한 소리를 인지하여 사람들에게 알릴 장치가 필요했다. 그것을 우리는 딥러닝을 통해 구현해 보기로 하고 ‘딥러닝 기반 위험 소리 감지 앱’이라 구상했다.

## 2. 연구 목표

### 가. 연구목적

‘딥러닝 기반 위험 소리 감지 앱’은 소리를 들을 수 없는 사용자에게 위험한 소리가 발생했을 때, 그 소리를 감지하여 소리 외 방법인 시각과 촉각으로 알려주는 것을 목적으로 한다.

### 나. 수행과정 및 방법

연구목적에 맞춰 사용자가 사용할 Client인 Application, 소리를 분석하여 나타낼 Server가 존재한다. Client에는 소리를 받아들이고 Server에 소리데이터를 보내며 Server는 받은 소리를 전처리하고 학습된 모델을 넘겨 무슨 소리

인지를 판단한다. 그 판단한 결과는 다시 Client로 넘어와서 사용자의 설정을 고려하여 그 결과를 나타낸다. 위의 과정은 백그라운드로 돌아가며 사용자는 설정만 해 두었을 때 소리가 나면 그 소리를 스마트폰의 화면 전체에 나타나는 색과 진동으로 무슨 소리가 났음을 인지할 수 있다.

#### 다. 연구 결과물 형태

해당 연구는 서버와 어플리케이션으로 구성되며, 그 중 사용자는 어플리케이션만을 결과물로 확인할 수 있다. 앱 설정을 통해 사용자의 환경에 맞춰 이용할 수 있게 될 것이다.

### 3. 역할 분담 및 팀 소개

소속	역 할	성 명	연락처
컴퓨터공학부	팀장, 설계, AI 모델링	김상우	010--0000
컴퓨터공학부	설계, AI 모델링	김문웅	
컴퓨터공학부	front-end, UI	김성원	
컴퓨터공학부	front-end	손영진	
컴퓨터공학부	back-end	차주원	

### 4. 연구 일정

연구내용 \ 월	1월	2월	3월	4월	5월	6월
인공지능 모델 탐색						
중간 발표 준비						
최종 발표 준비						
APP UI, Java 분야 구현						
외부 API 구현						
서버 인공지능 연동						
데이터베이스 구축						
인공지능 모델 학습						
테스트						
마무리						

## II. 본 론

### 1. 연구 내용

#### 가. 선행연구 및 제품 관련 자료조사·분석

##### 1) 딥러닝 모델

목표를 달성하기 위한 핵심부분은 딥러닝 모델을 구성하는 것이다. 입력데이터인 소리를 분석하기 위한 모델을 구현하고 파라미터 설정을 한다. 이후 정확도와 속도를 고려하여 최적의 모델을 선정하여 진행한다.

##### - CNN 모델

CNN 모델은 image를 분류하기 위해 개발된 Network이다. CNN 모델은 Convolution layer 와 Pooling layer의 조합으로 만들어진 모델들을 통틀어 CNN 모델이라 부른다.

여기서 Convolution layer는 이미지의 특징을 판단하는 layer이다. 그리고, Convolution layer는 이미지보다 작은 filter를 사용하여, filter로 특징을 판별한다. 즉 작은 필터로 이미지를 훑어보면서 연산하는 것이다.

다음은 Pooling layer인데, 이 layer는 특징을 강화하는데 목을 두고 있다. 사용되는 방식은 Convolution layer와 비슷하지만, 값들 중에서 특정 값만 유지하고, 나머지는 버리는 특징이 있다. Pooling layer 방식 중 Max Pooling 방식을 많이 사용한다. 이 방식의 장점으로는 연산의 부하가 적게 들고, 오버피팅을 약간 줄여주는 장점이 있다.

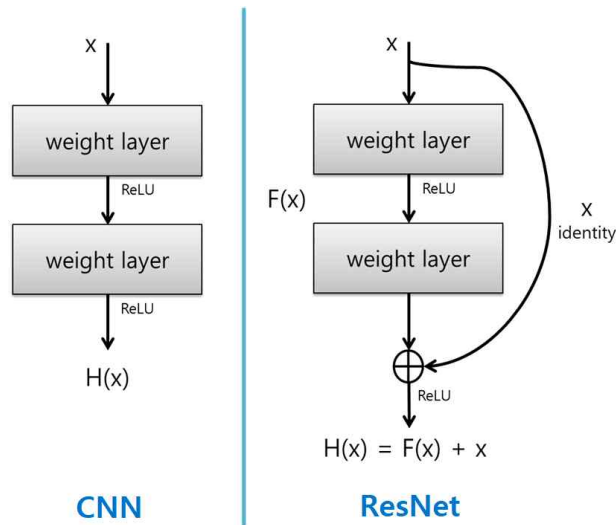
##### - LSTM 모델

LSTM 모델은 RNN 모델의 특별한 한 종류로, 긴 의존 기간을 필요로 하는 학습을 수행할 능력을 갖고 있다. 긴 의존 기간을 피하기 위해 LSTM은 기존 RNN구조에서 각 반복 모듈마다 다른 구조를 가지고 있게 되는데, 각 반복모듈마다 4개의 layer가 특별한 방식으로 정보를 주고받게 된다. 이것을 설명하자면, 먼저 첫 번째로 들어온 정보를 버릴 것인가를 정하는 단계가 1단계이고, 2번째 단계로는 들어온 정보 중 어떤 정보를 쓸 것인가를 정하게 된다. 3번째 단계에선 과거의 정보와 새로 들어온 정보를 합치는 단계를 진행한다. 마지막으로 어느 부분을 output으로 보낼 것인지를 정하게 된다.

이러한 LSTM 모델은 보통은 시계열 데이터를 학습을 하는데 사용된다.

### -ResNet CNN 구조

ResNet구조로 CNN 레이어를 구성했다. ResNet구조에선 Deep Residual Learning 방식을 쓰고있는데, 일반적인 방식과 DeepResidual Learning방식의 차이는 일반적인 딥러닝 네트워크는 쌓여진 layer들을 underlying mapping 하도록 (입력  $x$ 가layer를 통과한 output이 목표값  $y$ 에 딱 맞도록) 학습한다. Deep Residual Learning은 residual(잔차, 입력과 출력의 차이)를 mapping 하도록 학습합니다. 이 residual mapping을 위해 아래 그림과 같은 shortcut connection이라는 구조를 포함한 block을 설계한다.



오른쪽의 residual learning block의 구조를 살펴보면 왼쪽의 기존 네트워크 구조에서 하나 이상의 layer를 넘어서 연결하는 shortcut connection을 추가한 것을 볼 수 있다. shortcut connection은 입력 값을 layer들을 통과한 출력값에 더하며, 추가적인 파라미터가 없고 연산량 증가가 크지 않은 장점이 있다.

## 2) MFCC

음성 데이터의 MFCC(Mel-Frequency Cepstral Coefficient)를 뽑아내는 과정이다. 여기서 MFCC란 오디오 신호에서 추출할 수 있는 feature로, 소리의 고유한 특징을 나타내는 수치이다. 즉 소리 데이터를 벡터화 시키는 것이다.

MFCC를 뽑아내는 과정에는

1. Farming & Windowing
2. FFT
3. Mel Filter Bank
4. Cepstral Analysis

과정이 있다.

먼저 오디오 신호를 Farming & Windowing을 적용하는데, 오디오 신호를 프레임 별로 자른다는 의미이다.

그리고 프레임 별로 나누어진 오디오 신호에 FFT를 적용하여 주파수 영역의 표현이 가능해지고, Spectrum을 구하게 된다.

그리고 구해진 Spectrum을 Mel Spectrum으로 변환해야하는데, 그 이유는 Mel Spectrum이 사람의 특성을 반영하여 얻어내는 것이기 때문이다. 결국 듣는 것을 사람 기준으로 해야하기에, 사람의 특성을 반영한 Mel Spectrum을 사용하게 된다. 이 Mel Spectrum을 뽑아내는 과정이 Mel Filter Bank이다.

그리고 뽑아진 Mel Spectrum을 분석하게 되는데, 그 분석이 Cepstral Analysis이다. 즉, Spectrum에서 최대가 되는 지점들을 찾고, 그 지점들을 연결한 곡선을 Spectrum에서 분리해내는 것을 말한다. 분리하는 과정에서 도출되는 것이 MFCC이다. 그렇게 해서 소리의 특징을 가진 2차원의 벡터를 추출해낸 것이다.

### 3) Client (Application) 설계

#### - Recyclerview

리사이클러뷰(Recyclerview)는 리스트뷰(ListView)와 "사용 목적" 및 "동작 방식"이 매우 유사한 요소이다. 리사이클러 뷰는 아이템을 표시하기 위해 생성한 뷰를 재활용하여 리스트 뷰의 단점인 성능 저하를 보완한다. 리사이클러 뷰를 위한 구성요소는 다음과 같다.

#### 1. 리사이클러뷰 (Recyclerview)

v7 지원 라이브러리(v7 Support Library)에서 제공되는 위젯으로, 앞서 소개한 구성 요소들을 통해, 사용자 데이터를 리스트 형태로 화면에 표시하는 컨테이너 역할을 수행한다.

#### 2. 어댑터(Adpater)

리사이클러뷰에 표시될 아이템 뷰를 생성하는 역할이다.

#### 3. 레이아웃매니저(LayoutManger)

레이아웃매니저는 리사이클러뷰가 아이템을 화면에 표시할 때, 아이템 뷰들이 리사이클러뷰 내부에서 배치되는 형태를 관리하는 요소이다.

각 레이아웃매니저가 아이템 뷰들을 배치하는 형태는 다음과 같다.

- 1) 리니어(LinearLayout):수평(Horizontal) 또는 수직(Vertical) 방향, 일렬(Linear)로 아이템 뷰 배치.
- 2) 그리드(GridLayoutManager): 바둑판 모양의 격자(Grid) 형태로 아이템 뷰 배치.
- 3) 스테거드그리드(StaggeredGridLayoutManager): 엇갈림(Staggered) 격자(Grid) 형태로 아이템 뷰 배치.

- 외부라이브러리: ColorPickerView(색상 선택창)

해당 프로젝트의 기능 중 하나인 색상 변경 기능을 구현하기 위하여 원하는 색으로 선택할 수 있도록 하는 색상 선택창인 ColorPicker 외부 라이브러리를 사용하였다.

해당 라이브러리는 색상 팔레트를 선택하여 사용자가 원하는 색상을 선택할 수 있다. 또한 라이브러리에서 제공하는 BrightnessSlider 액션바를 사용하여 사용자가 원하는 대로 색상의 밝기를 조절할 수 있으며, AlphaSlideBar 액션바를 alpha값을 수정하여 색상의 투명도 또한 조절이 가능하다.

- setContentView()

1개의 소리가 감지되었을 때는 하나의 소리만을 나타내도록 하였다. 하지만 위험한 소리가 2개 감지했을 때, 하나의 소리만을 표현하기에는 부족하다고 판단하여 setContentView() 함수를 활용하여 2개의 소리를 표현할 수 있는 XML 레이아웃으로 변경하여 2개의 소리 모두 사용자가 인식할 수 있도록 한다.

- 백그라운드

앱 화면이 꺼지거나, 앱이 꺼졌을 때 서비스를 이용하고 싶다면 백그라운드 실행 버튼을 통해 백그라운드로 소리를 녹음하고 결과를 알림으로 알려주게끔 구현했다. 안드로이드 스튜디오에 Service 클래스를 사용하여, 앱이 종료되어도 서비스는 계속 실행할 수 있도록 구현했다.



#### 4) Server 설계

이번 프로젝트에서 백엔드의 역할은 음성데이터의 전처리와 전처리된 데이터를 어플리케이션 쪽으로 보내는 것이다. 백엔드 프레임워크는 파이썬 기반의 Flask를 사용하였다.

서버는 아마존에서 제공하는 AWS EC2 인스턴스를 이용하여 배포하였다. AWS 서비스가 끊기더라도 서버가 중단 없이 돌아가게 하게끔 PM2를 이용하여 서버배포를 유지할 예정이다.

### 나. 작품 특징

이 작품에는 서버, 앱으로 구성되어 있으며, 사용자는 앱의 백그라운드 기능을 통해 소리를 인지할 수 있게 된다. 서버와 앱에는 각각의 전처리 과정과 AI모델을 가지고 있고 흐름은 다음과 같다.

앱에서는 들어온 소리를 짧은 시간으로 나눠 서버에 데이터를 송신하게 된다. 이후 자신 내부의 전처리 과정을 걸쳐 AI모델(A)를 걸치도록 한다. A 모델은 소리를 처리해야 할 소리와 처리하지 않아도 될 잡음으로 판단하는 모델로 모델결과로 binary 값을 출력한다. 모델의 결과값을 바탕으로 잡음일 경우 소리에 대한 피드백을 하지 않으며, 그 외 처리해야 되는 소리의 경우 서버에 결과를 요청하고 받은 결과와 설정을 반영하여 소리에 대한 피드백을 진행한다.

서버에서는 앱에서 받은 소리데이터를 전처리하여 가지고 있는 AI모델(B)를 거치도록 한다. B 모델은 소리데이터를 6가지 소리 중 하나로 분류하는 모델이다. 이 모델의 출력은 argmax화 되어 소리 간 비율로 나타나진 배열로 나타난다. 이후 앱에 결과를 송신하면서, 데이터를 삭제해 메모리를 비운다.

앱에서는 받은 데이터와 설정을 반영하여 모바일에 진동과 색으로 소리에 대한 피드백을 제공한다.

### 다. 기대효과

청각장애인에게는 위험이 닥치기 전 인지하고 회피할 수 있는 기회를 제공할

수 있으며, 노이즈캔슬링을 사용하고 있는 일반인에게도 위급상황에서의 상황인지를 할 수 있다. 기회를 바탕으로 사고를 피해 사고율이 낮아지는 효과를 기대할 수 있다.

## 2. 문제점 및 해결방안

### 가. 문제 인식

(1)

사용자가 색상을 직접 선택할 수 있으며, 밝기와 투명도 또한 조절할 수 있어야 한다. 이후에 나올 스마트폰과의 호환성에 문제가 없어야 한다.

(2)

위험한 상황이 2가지가 아닌 1가지만 존재할 때도 메인 화면에 위험 상황을 2개 알려준다.

위험한 상황이 1가지인 경우 사용자가 혼란을 겪어서는 안된다.

(3)

기존에 CNN 모델 구조가 단순하다는 평가를 받아 모델 구조를 새롭게 만들어 Classification을 좀 더 정확하게 해야 한다.

(4)

Validation DataSet에서는 좋은 지표가 나왔지만 실제 테스트에서는 Validation Dataset에서 만큼의 지표가 나오지 않았다. 그렇기에 실제 테스트에서도 그만큼의 성능까지 끌어올려야 한다.

### 나. 문제 정의

(1)

문제 정의:

색상 선택 기능을 구현할 수 있는 외부 라이브러리들을 파악하고, 최근까지 commit이 활발하고, 밝기와 투명도를 조절가능한 라이브러리를 선정하여야 한다.

#### 해결 방안:

최근까지 commit이 활발한 skydoves/ColorPickerView 라이브러리를 프로젝트에 추가하여 기능을 구현한다.

```
//color picker  
implementation "com.github.skydoves:colorpickerview:2.2.4"
```

(2)

#### 문제 정의:

위험한 소리가 2개라고 판단할 경우 2개를 표현하고, 1개일 경우 1개만을 표현하도록 하여 사용자에게 혼란을 주지 않도록 한다.

#### 해결 방안:

setContentView()함수를 사용하여 1개만 표현하는 레이아웃을 생성하여 위험 소리가 1개라고 판단할 경우 해당 레이아웃을 출력하도록 구현한다.

(3)

#### 문제 정의:

기존의 CNN 모델 구조가 단순하기에 좀 더 효과적인 CNN 모델 구조를 파악하고, 우리 프로젝트와 맞는 모델 구조를 찾아야 한다.

#### 해결 방안:

CNN 모델 구조를 만들 때, 가장 대표적으로 ResNet 구조와 GoogleNet 구조를 많이 사용한다는 것을 알았고, 그중 ResNet 구조가 테스트를 해봤을 때 정확도 면이나 구현 면에서 좋다는 결과를 얻어 ResNet 구조로 CNN 모델을 만들었다.

(4)

#### 문제 정의:

Validation Dataset에서와 실제 테스트에서의 차이를 알아내고, 방법을 생각

해내야 한다.

#### 해결 방안:

학습시킬 때 데이터 안에 불필요한 소리들이 섞여있는 것을 확인하여 원본 데이터들을 각 라벨에 맞는 소리만 들어가게끔 편집하여 데이터를 가공하였다.

다. 현실적 제한 조건

#### <현실적 제한조건 설명>

현실적 제한조건	규격/표준	O	- 어플리케이션 제작 제한에 고려하여 개발 - 갤럭시 계열 스마트폰의 기종 간 호환성 고려
	경제성	O	- 추가적인 비용 없이 접근성이 편한 앱 사용
	미학	O	- 소리가 필요 없는 UX/UI 고려 - 간단한 디자인으로 필요한 역할을 진행하도록 설정
	신뢰성/내구성	O	- 백그라운드에서 돌아갈 때의 앱 자원 소모 고려 - 서버 통신 시 조건을 부여해 부하 감소
	안전성	O	- 우선순위 설정 시 설정에 따른 영향 공지
	윤리/환경	O	- SW관련 규정 준수(오픈소스 및 API 불법 사용 유무 검토)

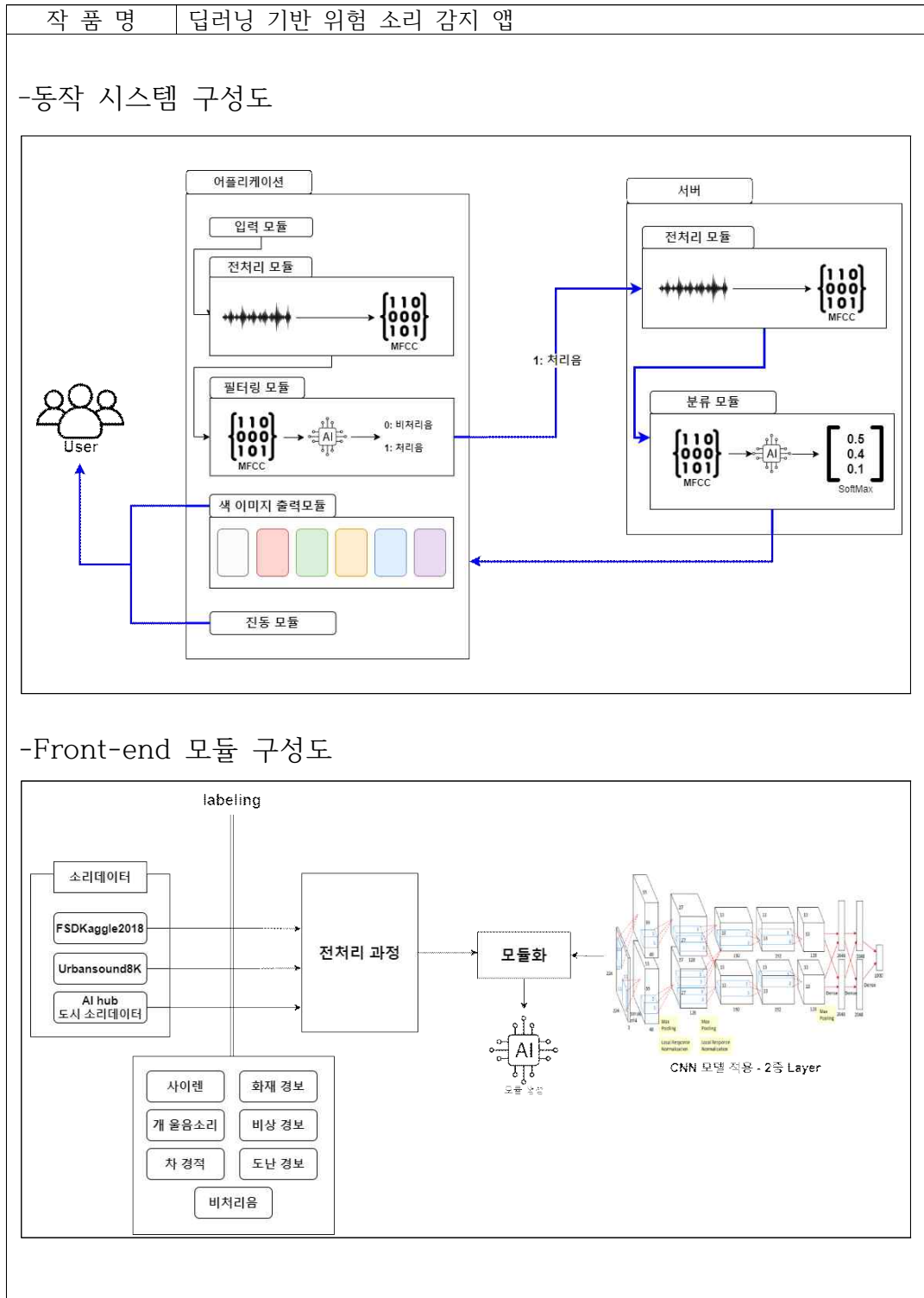
### 3. 공학문제수준설명

※ ‘속성1’ 필수, ‘속성2~8’ 중에 4개 이상 선택

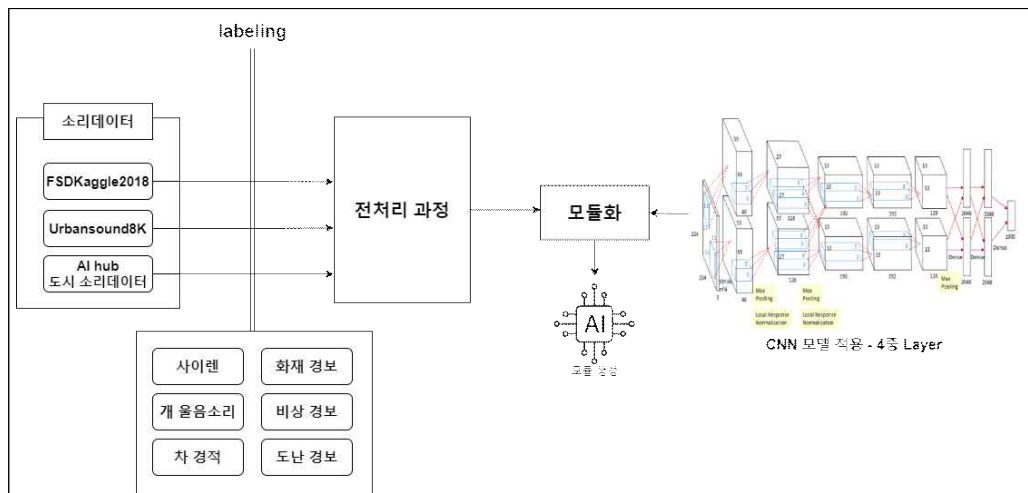
문제의 속성	공학문제수준설명	학생의견	
	심화된 공학문제가 속성1(지식의 깊이)을 만족하고, 속성2~속성8 중 일부 또는 전부를 만족해야 한다.	만족여부 (O/X)	만족사유
속성1 (지식의 깊이)	최신 정보와 관련 연구 결과를 활용하고 있다.	O	최근 이슈가 많은 AI 와 관련된 주제를 선정했다.
속성2 (상충되는 요건의 범위)	상충될 수 있는 기술적 또는 공학적 이슈를 다루고 있다.	O	소리와 이미지라는 다른 속성을 변환하는 전처리 과정을 진행한다.
속성3 (분석의 깊이)	해답이 명확하지 않은 문제를 해결하기 위해 깊이 있는 사고와 분석과정을 다루고 있다.	O	장애를 보완하는 방법에 정답이라고 할 것은 없으나, 현 환경에서 비용적, 실사용적 시선에서 분석하여 주제를 선정하였다.
속성4 (생소한 주제)	자주 접하지 않는 공학문제를 다루고 있다.		
속성5 (문제의 범위)	전공분야의 일반적인 실무 영역을 벗어난 범위를 다루고 있다.	O	소리를 자연어처리하지 않고 이미지처럼 처리하는 기법을 사용하였다.
속성6 (이해당사자의 요구 수준 및 범위)	다양한 이해당사자들의 요구사항들을 고려하고 있다.	O	최대한 간단하며 단순화된 사용법으로 누구나 사용할 수 있도록 하였다.
속성7 (상호의존성)	상호 의존적인 여러 세부문제들이 결합된 종합적인 문제로 구성되어 있다.		
속성8 (다양한 영향 고려)	다양한 분야에 미치는 영향을 고려하고 있다.		

### Ⅲ. 결론

#### 1. 연구 결과



## -Back-end 모듈 구성도

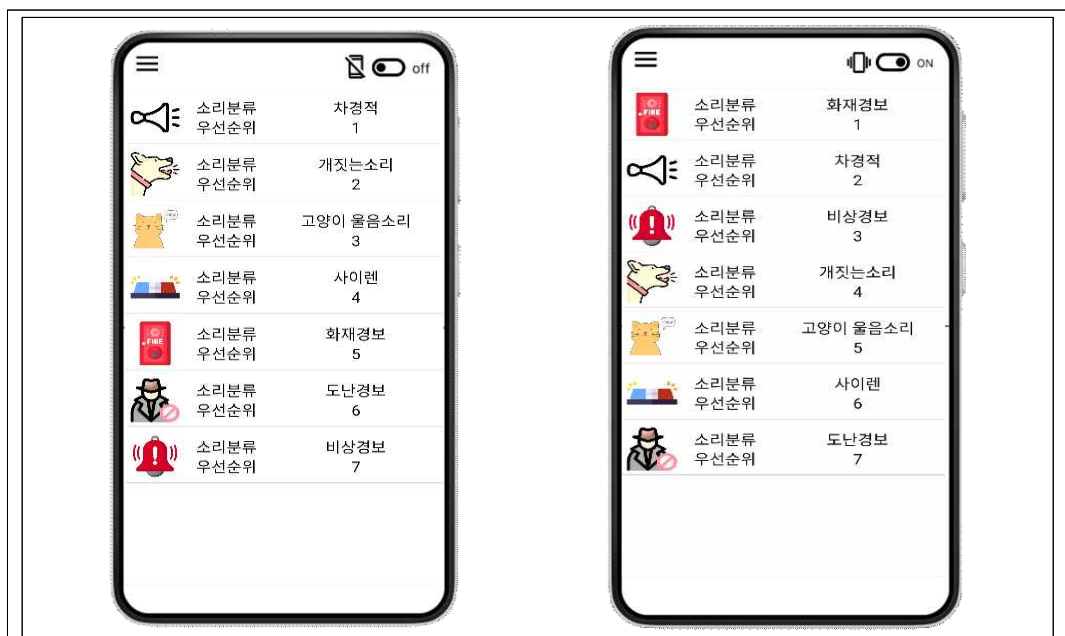


## - AI 모델 비교 (F1 Score)

	LSTM	CNN	ResNet
F1-Score			0.9490
Accuracy			97

## - 구현된 화면





### 색상 설정 페이지





## 2. 작품제작 소요재료 목록

제품명	용도
서버	AI 개발
스마트폰	하드웨어
파이썬	AI 개발
헤드셋	테스트
텐서플로우	AI 개발

## 참고자료

- 1) 서지영, 딥러닝 파이토치 교과서, (주)도서출판길벗, 2022
- 2) 리사이클러뷰, <https://recipes4dev.tistory.com/154>
- 3) <https://github.com/skydoves/ColorPickerView>