

졸업 작품 보고서



Smart Homecam

2022년 12월

서경대학교 컴퓨터공학과

이현수

목 차

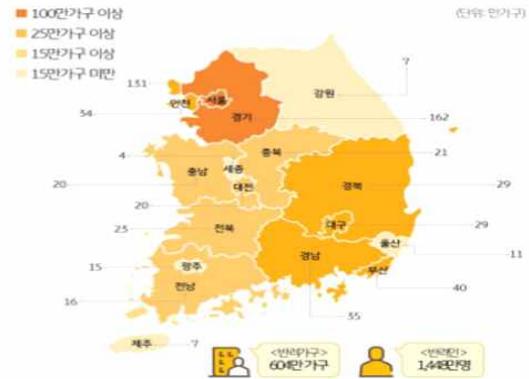
I . 서론	2
1. 제작 동기 및 목적	2
2. 제작 내용	3
II . 관련 연구 및 동향	4
1. 홈캠 시장 동향	4
III . 개발 내용	5
1. 마이페이지	7
2. 실시간 스트리밍	8
3. 캡처 및 동영상 녹화	9
4. 홈카메라 모드	13
(1) 사람 탐지 모드	14
(2) 화재 탐지 모드	16
(3) 외부인 탐지 모드	18
(4) 반려동물 탐지 모드	21
(5) 안심 모드	23
5. 이메일 · SMS 알림	25
6. AWS 클라우드 배포	26
IV . 개발 결과	28
V . 결론	29

I. 서론

1. 제작 동기 및 목적

1인 가구 추이 (단위: 만 가구)

자료: 통계청



1인 가구 추이와 반려가구 통계

‘2021년 인구주택총조사 결과’에 따르면, 지난해 기준 1인 가구는 716만6000가구에 집계됐다. 이는 전체 가구 중 33.4%에 해당한다. 2020년 말 현재 반려동물을 기르는 ‘반려가구’는 604만가구에 한국 전체 가구의 29.7%를 차지하고, 반려인은 1,448만 명으로 한국인 4명 중 1명 이상이 반려동물과 함께 살아가고 있다. 이는 통계청, 농림축산식품부 동물등록정보 그리고 전국 20세 이상 남녀 1,000명을 대상으로 실시한 설문조사 결과를 기초 자료로 활용해 추정한 수치다.

2021년 KB금융그룹의 한국 반려동물 보고서에 따르면, 반려동물이 혼자 집에 있는 시간은 하루 평균 5시간 40분이라고 합니다. 특히, 1인 가구의 경우 반려동물이 혼자 보내는 시간이 평균 7시간 20분으로 더 길었다. 직장을 다니는 1인 가구라면 최소 9시간 이상 반려동물을 혼자 두는 경우가 대부분일 거라고 예상된다.

1인 가구의 경우 집을 비울 때 범죄, 화재 발생 등의 사고에 대해 민첩하게 대응하지 못한다. 집에 동거인이 없는 경우 침입 범죄가 발생한다면 더 큰 문제가 발생할 수 있고, 화재의 경우 기존 화재 정보 시스템은 온도 기반이므로 빠른 대처가 불가능하다. 또 최근 독거노인의 고독사 문제는 꾸준히 증가하는 추세이다. 또 1인 가구의 경우 반려동물이 혼자 지내는 시간이 늘어남에 따라 반려동물의 건강 상태를 확인할 수 없었다. 이러한 사회 문제를 해결하기 위해 ‘SmartHomecam’ 작품을 개발하게 되었다.

‘SmartHomecam’ 작품을 통해서 가정 내의 신속한 범죄 인지와 대처, 빠른 화재 감지로 인명/재산 피해 최소화, 고독사 문제 해결, 반려동물 모니터링을 할 수 있다. 이렇게 함으로써 가정 내에서 발생하는 각종 사고를 빠르게 인지하고, 가정의 안전을 보호할 수 있을 것으로 기대된다.

2. 제작 내용

- HW

라즈베리파이, 오디세이 보드를 사용해 카메라 영상 프레임을 서버로 실시간으로 전송한다.

- 홈카메라 실시간 모니터링

서버에 연결된 임베디드 보드의 카메라 영상을 실시간 모니터링할 수 있고, 동시에 이미지 캡처, 영상 녹화가 가능하다.

- 홈카메라 모드

사람탐지, 화재탐지, 외부인탐지, 반려동물 탐지, 안심모드(일정시간 사람 미탐지, 행동 미감지) 모드가 있다.

- 마이페이지

가족 구성원 관리, 홈카메라 모드 설정, 알람 확인, 통계 확인, 탐지 기록 확인을 할 수 있다.

- 이메일, SMS 알람

반려동물 탐지를 제외한 홈카메라 모드에 대해서 특정 상황 발생 시 등록된 이메일, 전화번호로 알람이 전송된다. 이때 구글을 이용해 이메일을 전송하고, AWS SNS 서비스를 이용해 SMS 메시지를 전송한다.

- AWS 클라우드 배포

AWS EC2, AWS RDS, AWS S3 Bucket을 사용해 클라우드에 배포했다. 데이터는 AWS RDS에 저장되고, 사용자 저장 이미지 캡처, 녹화 동영상, 가족 구성원 얼굴 사진, 홈카메라 모드 탐지 이미지는 AWS S3에 저장된다. 배포 시 docker-compose를 사용해 Django(AWS)와 Nginx(Web server)를 배포했다. 그리고 Python WSGI인 Gunicorn을 사용해 Nginx로부터의 서버사이드 요청을 처리했다.

II. 관련 연구 및 동향

1. 홈캠 시장 동향

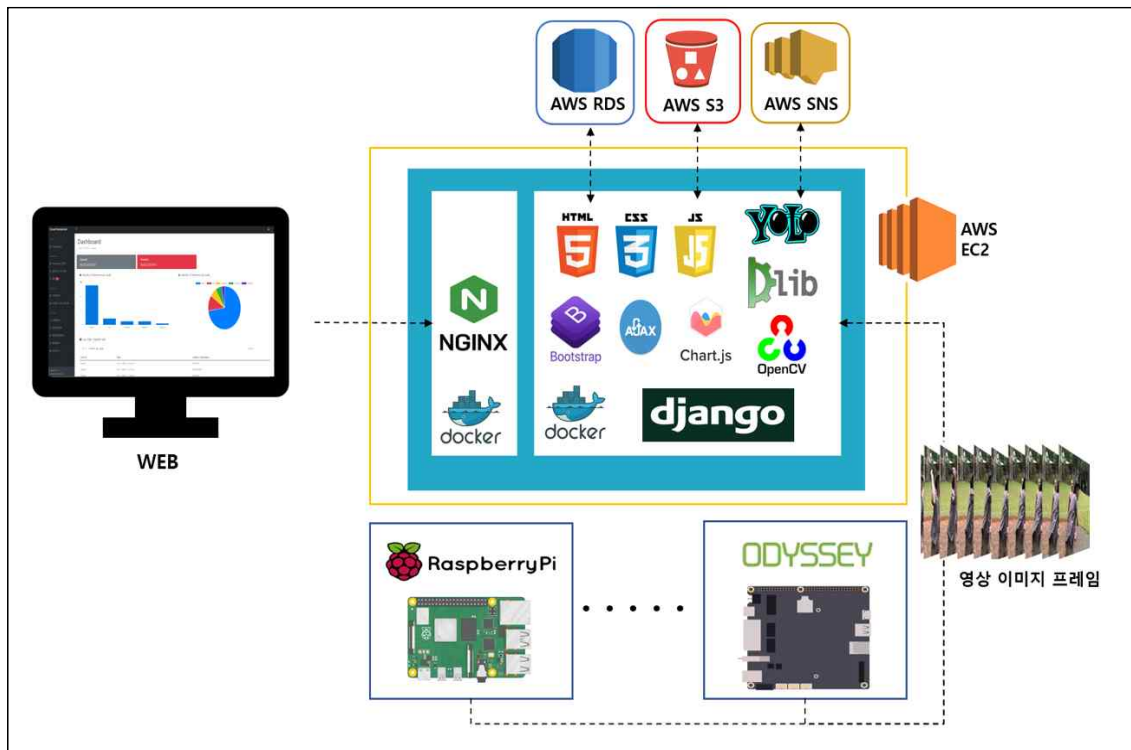
최근 1인 가구가 증가하고, 아파트 등 공동주택을 출입할 때 발생하는 사건·사고가 빈번해지면서 '스마트 홈 보안' 상품을 이용하려는 소비자들이 늘고 있다. 코로나19로 급증한 각종 배달과 택배 사고에 대한 염려가 커진 영향이 크다. 여성이 집에 혼자 있을 때, 또는 직장인 여성들이 집에 혼자 있는 자녀들이 배달음식 등 낯선 사람의 방문을 받을 때를 대비해 사용할 수 있는 서비스에 대한 수요가 많아진 셈이다. 특히 클라우드 저장공간과 AI 기술을 접목하면서 가정용 보안 서비스를 사용할 때 가장 큰 심리적 장벽인 가격에 대한 부담을 줄인 서비스들도 나오고 있다. '보안 시스템은 고급주택의 전유물'이라는 인식을 깨기 위해 마련된 홈 보안 서비스들이다.

ADT캡스는 아파트에서도 쓸 수 있도록 '현관문 앞 보안'에 집중한 월 1만원대 구독형 서비스 '캡스홈 도어가드'를 선보여 빠른 성장세를 보이고 있다. '캡스홈 도어가드'는 현관문에 부착하는 CCTV와 모바일 앱이 연동돼 현관 앞 상황을 언제 어디서나 실시간 영상으로 확인할 수 있도록 했다. 현관 앞에 방문자나 배회자가 있는 경우 자동으로 감지해 사용자 스마트폰 앱으로 알림과 영상을 보낸다. 주 이용 고객은 싱글 여성과 3040 부모들이다. 3040 세대가 전체의 70% 가량을 차지하고, 여성이 전체의 65%다.

LG유플러스는 '우리집 지킴이 이지(Easy)' 라는 패키지를 월 8800원에 제공한다. 실내 영상을 찍어주는 '맘카 이지'와 동작감지센서, 도어센서와 KB손해보험 서비스를 결합한 상품이다.

샤오미는 '미 스마트홈캠 360°'을 국내 출시했다. 가정용 보안 감시 카메라로 24시간 집안 전체를 360° 커버해 감시할 수 있다. AI 인체감지 알고리즘을 채택해 사람 실루엣을 감지하고 스마트폰 미홈 앱을 통해 알아서 알림을 보낸다. 이 경우 자동 얼굴인식 기능이 탑재돼 사전 등록된 가족·지인 등의 얼굴 식별이 가능하다.

Ⅲ. 개발 내용

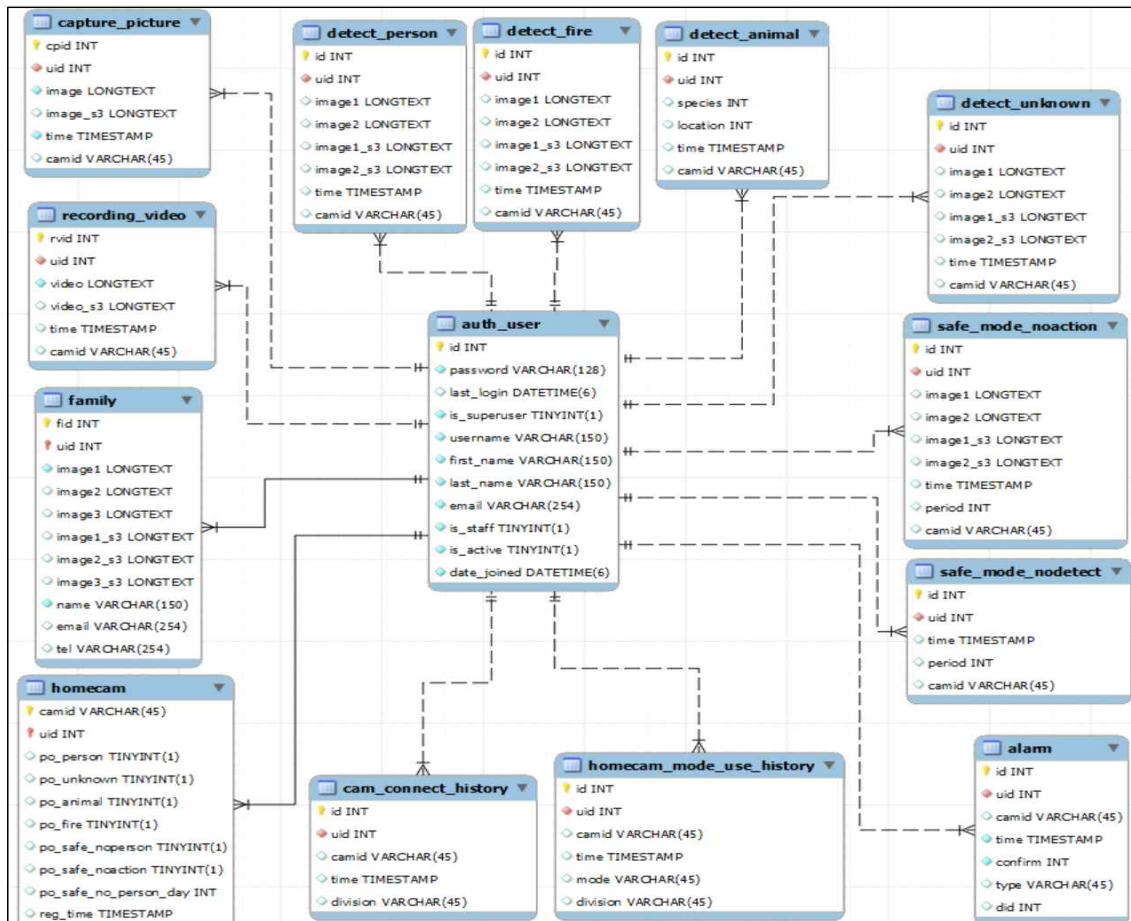


시스템 구성도

본 프로젝트는 임베디드 시스템과 인공지능을 이용한 줄음감지 웹 시스템이다. 카메라가 부착된 임베디드 보드가 홈카메라 역할을 한다. 여기서 홈카메라는 서버에 이미지 프레임을 전송하는 역할만 한다. 그 후 서버에서는 인공지능 알고리즘을 이용해 알고리즘에 정의된 상황을 감지하고 사용자에게 Web, Email, SMS를 통해 알람을 전송하고 상황을 확인할 수 있다.



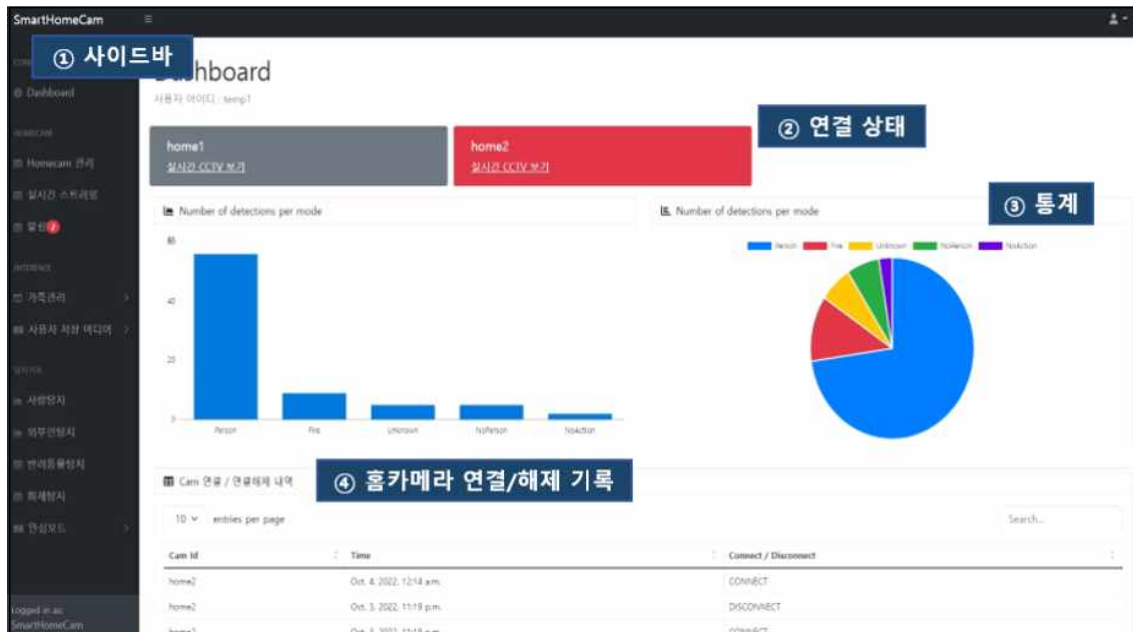
라즈베리파이4 8GB(좌), 오디세이 보드(우)



데이터베이스 ERD

- auth_user : 사용자 정보
- homecam : 사용자 계정에 연결된 홈카메라 정보
- family : 가구원 등록 정보
- capture_picture : 사용자 캡처 저장 정보
- recording_video : 사용자 녹화 영상 저장 정보
- homecam_mode_use_history : 홈카메라 5가지 모드 on, off 기록
- cam_connect_history : 라즈베리파이 연결, 해제 기록
- alarm : 알람 저장
- detect_person : 홈카메라 사람탐지모드 저장
- detect_aniaml : 홈카메라 반려동물탐지 모드 저장
- detect_fire : 홈카메라 화재탐지 모드 저장
- detect_unknown : 홈카메라 외부인 탐지 모드 저장
- safe_mode_nodetect : 홈카메라 안심모드 사람 미탐지 저장
- safe_mode_noaction : 홈카메라 안심모드 사람 행동 미감지 저장

1. 마이페이지



마이페이지 내 Dashboard

본 프로젝트는 웹기반의 서비스이다. 마이페이지를 통해서 홈카메라 관리, 실시간 스트리밍, 알림 확인, 탐지 내역 확인, 모드 on/off 기능 등 모든 기능을 사용하고 제어할 수 있다. 위 사진은 마이페이지 내 Dashboard이다.

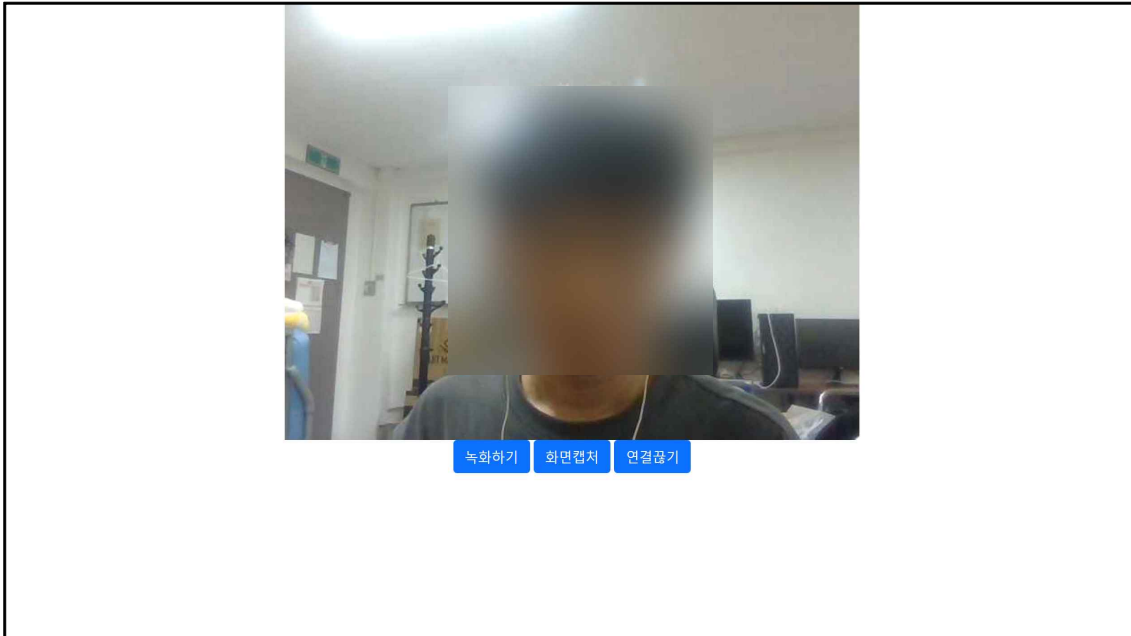
① 사이드바는 사이드바를 통해 다른 기능 페이지로 이동할 수 있다. Homecam 관리 페이지, 가족 구성원 관리, 사용자 저장 미디어 확인, 모드 별 탐지 기록 확인 페이지로 이동할 수 있다.

②연결상태는 DB에 등록된 홈카메라를 연결 상태에 따라 변하는 색깔을 통해 시각적으로 현재 홈카메라의 상태를 확인할 수 있다. 만약 특정 홈카메라가 연결되면 회색->파란색 혹은 회색->빨간색으로 변경된다. 만약 미확인 알림이 있을 경우에 회색->빨간색으로 변하고, 그것이 아닐 경우 회색->파란색으로 변한다. 이는 Ajax 비동기 통신을 이용해 1초마다 서버에 지속적으로 해당 홈카메라의 연결 상태를 확인해 구현했다.

③ 통계는 모드별 탐지 수를 그래프로 시각화해서 보여준다. 반려동물 탐지를 제외한 사람/화재/외부인/행동미감지/사람미감지 탐지 수를 시각적으로 볼 수 있다.

④ 홈카메라 연결/해제 기록은 홈카메라의 연결 시간과 해제 시간을 표 형태로 보여준다.

2. 실시간 스트리밍



실시간 스트리밍 페이지

실시간 스트리밍 페이지에서는 임베디드 기기에서 전송되는 영상 프레임을 전송받아 사용자가 실시간으로 확인할 수 있다. 이 페이지에서는 부가적으로 동영상 녹화, 이미지 캡처, 연결해제 버튼이 존재해 해당 기능을 이용할 수 있다.

임베디드 기기와 TCP 소켓 통신을 이용해 통신한다. 서버에서는 다중 사용자의 다중 홈카메라를 연결해야 하기 때문에 홈카메라를 연결/관리하는 VideoCamera 객체는 싱글톤으로 만들었다. 그 후 다중소켓통신과 멀티스레드 프로그래밍을 이용해 다중 사용자 / 다중 홈카메라의 연결을 가능하게 했다.

임베디드 기기에 연결된 카메라 프레임을 얻기 위해 opencv 모듈의 VideoCapture 메서드를 이용한다. TCP 소켓 연결 후 username과 홈카메라 id를 전송한다. 그 후 네트워크 부하를 줄이기 위해 이미지 품질을 50% 감소시키고 pickle 모듈을 사용해 직렬화 한다. struct.pack을 사용해 바이트 객체로 변환한다. 빅 엔디안(>, 최상위 바이트부터 차례대로 저장하는 방식)과 unsigned long long(Q) 크기로 영상 프레임을 전송한다. 이때 로컬 환경에서는 빅 엔디안 방식과 unsigned long(L) 크기로 정상적으로 전송이 되었지만, 클라우드 환경에서는 빅 엔디안 unsigned long long(Q) 크기에서 정상적으로 동작했다.

실시간 스트리밍 페이지에서 전송받은 영상 프레임을 사용자에게 보여줄 때, StreamingHttpResponse을 사용했다. 로컬에서 정상 동작했지만, 클라우드 환경에서 동작하지 않는 문제가 있었지만 gunicorn의 threads 옵션을 사용해 해결했다.

3. 캡처 및 동영상 녹화



동영상 녹화 / 이미지 캡처

① 실시간 스트리밍 기능은 해당 페이지에서 보여지는 화면은 임베디드 기기에서 실시간으로 전송되는 영상이다.

② 녹화하기, ③ 화면캡처, ④ 연결끊기 버튼의 경우 Ajax 통신을 이용해 구현했기 때문에 해당 기능을 이용하기 위해 버튼을 클릭하더라도 페이지 새로고침 없이 해당 기능을 동작한다.

② 녹화하기는 실시간 스트리밍 페이지에서 보여지는 화면을 녹화하는 기능이다. 녹화하기 버튼을 누르면 “녹화하기” 버튼이 “녹화중”으로 변경된다. 여기서 “녹화중” 버튼을 클릭하면 “녹화하기” 버튼으로 변경되고 영상은 AWS S3 Bucket에 저장된다. ③ 화면캡처는 현재 실시간 스트리밍 페이지에서 보여지는 화면을 순간 이미지로 캡처해 AWS S3 Bucket에 저장하는 기능이다.

[Amazon S3](#) > [바킷](#) > [us-east-1>awslogs-1-zzz-1-111111111111-us-east-1>awslogs-1-zzz-1-111111111111-us-east-1>recording/](#)

recording/

S3 URI 복사

역제
속성

객체 (11)

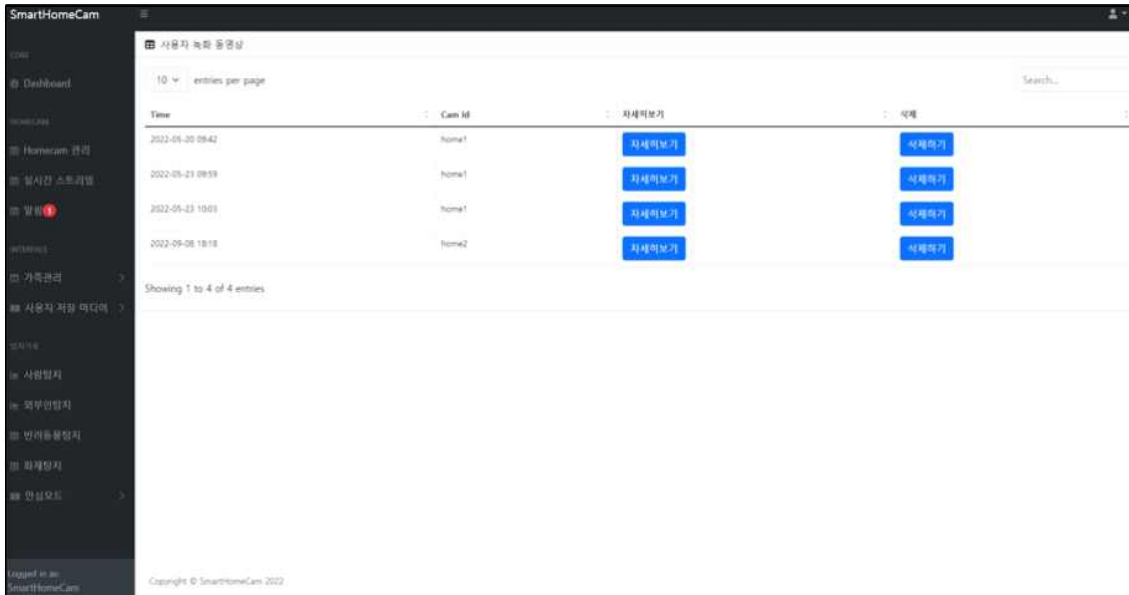
객체는 Amazon S3에 저장되어 있는 기본 엔티티입니다. Amazon S3 인벤토리를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. [자세히 알아보기](#)

S3 URI 복사
URL 복사
다운로드
열기
삭제
작업 ▼
폴더 만들기
업로드

합두사로 객체 찾기

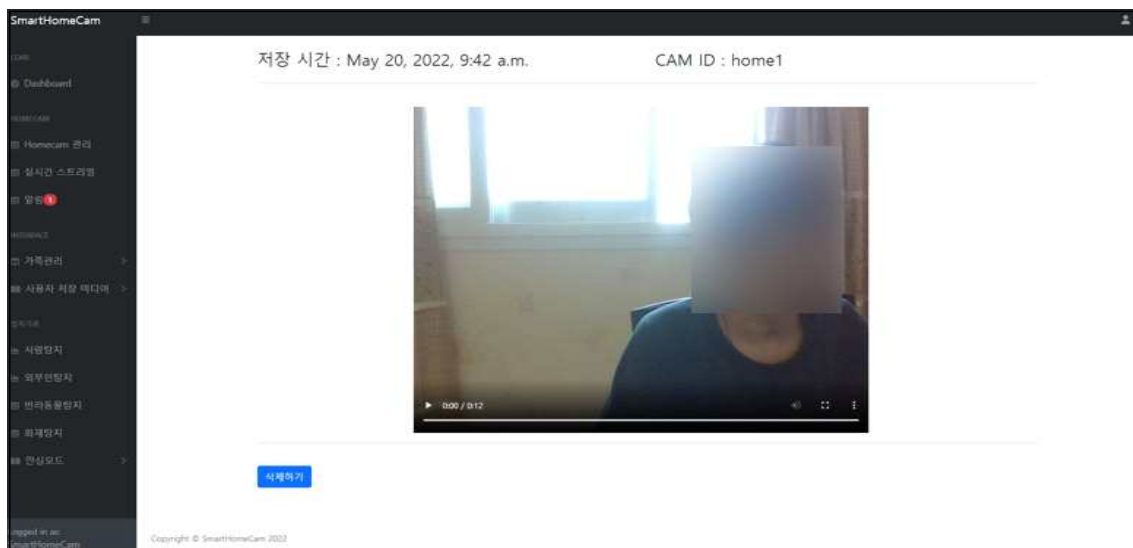
<input type="checkbox"/>	이름	▲ 유형 ▼	마지막 수정 ▼	크기 ▼	스토리지 클래스 ▼
<input type="checkbox"/>	O21d60af-f106-48a7-802d-05bcc11bbad3.mp4	mp4	2022. 10. 11. pm 2:19:29 PM KST	1.7MB	Standard
<input type="checkbox"/>	Qada165a-c8e9-4bed-902f-559e76dd5166.mp4	mp4	2022. 11. 11. pm 3:34:51 PM KST	410.7KB	Standard
<input type="checkbox"/>	37d83e2d-0513-4ca0-911a-e09820B06ead.mp4	mp4	2022. 10. 11. pm 2:34:14 PM KST	1.1MB	Standard
<input type="checkbox"/>	41523e3e-5599-4927-a54f-8a2b001b3ba6.mp4	mp4	2022. 11. 11. pm 2:26:08 PM KST	1.1MB	Standard
<input type="checkbox"/>	63873537-7e61-4b65-8796-36b062959c6c.mp4	mp4	2022. 10. 3. pm 11:19:38 PM KST	12.2MB	Standard
<input type="checkbox"/>	7hb51bha-676b-4dd7-b09d-38b4bba66661.mp4	mp4	2022. 10. 5. am 12:11:50 AM KST	2.4MB	Standard
<input type="checkbox"/>	aa2b50fc-0642-441f-98d5-4098612a35d2.mp4	mp4	2022. 10. 11. pm 2:17:00 PM KST	5.3MB	Standard
<input type="checkbox"/>	b13242bd-7549-4887-8f4b-e5b6a51bf9d7.mp4	mp4	2022. 10. 27. am 8:49:19 AM KST	1.1MB	Standard
<input type="checkbox"/>	bd9db31e-20fc-4976-915f-beeb46e2a39.mp4	mp4	2022. 10. 11. pm 2:38:38 PM KST	2.8MB	Standard
<input type="checkbox"/>	e0d11656-09b8-43b3-9d0c-6e9d52cf54.mp4	mp4	2022. 10. 11. pm 2:47:40 PM KST	15.8MB	Standard
<input type="checkbox"/>	eeb011e9-e806-4f8c-b07e-e2fa808d3be3.mp4	mp4	2022. 10. 5. am 12:14:06 AM KST	4.2MB	Standard

AWS S3 Bucket에 저장된 동영상 파일



사용자 저장 미디어 - 동영상 페이지

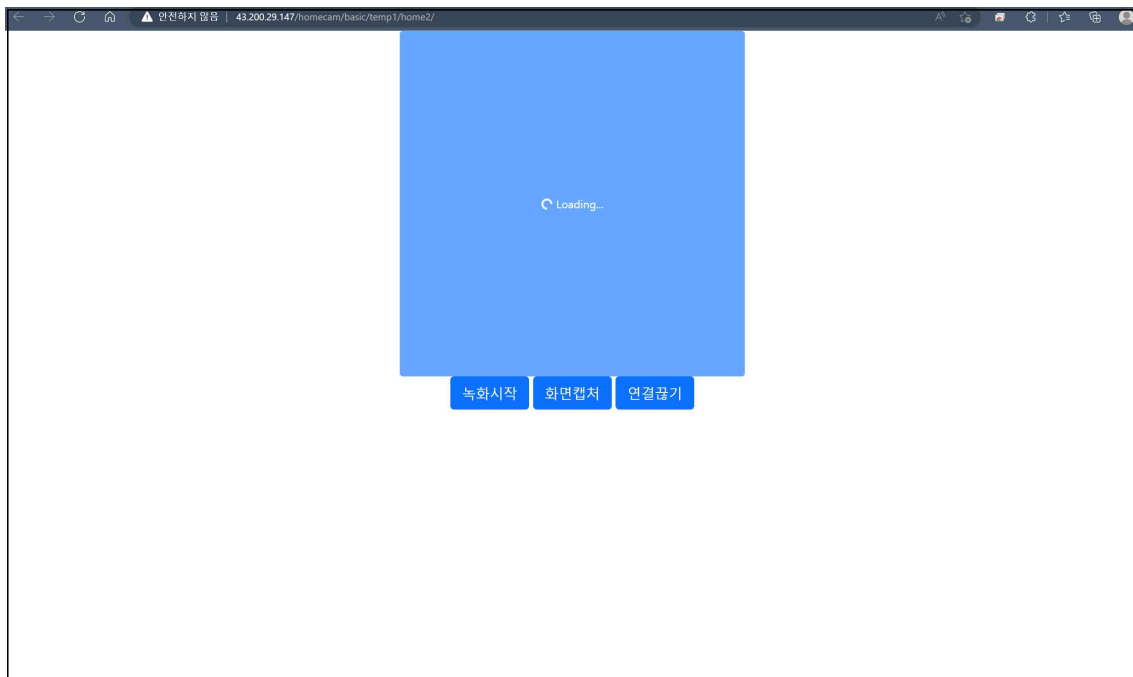
사용자가 저장한 미디어 파일(동영상/이미지)은 마이페이지 사이드바의 사용자 저장 미디어 탭을 이용해 확인할 수 있다.



저장된 동영상 확인

자세히보기 버튼을 클릭하면 위 사진처럼 동영상/이미지를 볼 수 있고 삭제하기 버튼을 클릭하면 삭제된다. 동영상/이미지의 경우 AWS S3 Bucket에 저장되어 있기 때문에 html img/video 태그의 src 속성에 해당 미디어 파일 url을 넣는 방식으로 구현했다.

④ 연결끊기는 현재 실시간 스트리밍을 제공하는 홈카메라의 소켓 통신을 해제하는 기능이다. 4번 연결끊기 기능은 현재 보고 있는 홈카메라와 서버의 소켓 연결을 해제하는 기능이다. 연결 해제 후 재연결이 가능하다.



Ajax 통신을 이용한 홈카메라 연결 여부 확인

② 녹화하기 기능은 Ajax를 통해 서버에 영상 녹화를 요청한다. 따라서 실시간 스트리밍 페이지를 닫아도 영상녹화를 중지하지 않는이상 계속해서 영상은 녹화된다. 실시간 스트리밍 페이지에서는 StreamingHttpResponse과 동영상 녹화 여부를 위한 Ajax 통신과 해당 홈카메라 연결 유무를 확인하는 ajax 통신이 동작한다. 따라서 로컬 환경에서는 문제가 없었지만, nginx 서버를 사용하는 경우 여러 스레드의 요청을 처리하지 못해 클라우드 환경에서는 gunicorn의 threads - 4 옵션을 사용해 해야 실시간 스트리밍 화면과 Ajax 통신을 기반으로 하는 동영상 녹화 / 이미지 캡처 / 연결 해제 요청을 정상적으로 처리할 수 있다.

```
stream mapping:
Stream #0:0 -> #0:0 (mpeg4 (native) -> h264 (libx264))
Press [q] to stop, [?] for help
libx264 @ 0x53f8c2295000 using CABAC=1
libx264 @ 0x53f8c2295000 using cpu capabilities: MMX2 SSE2Fast SSE3 SSE4.2 AVX FMA3 BMI2 AVX2
libx264 @ 0x53f8c2295000 profile skip, level 3.0, 4220, 8-bit
libx264 @ 0x53f8c2295000 264 - core 160 r301 cde9a93 - H.264/MPEG-4 AVC codec - Copyleft 2003-2020 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex submm=7 p
cpu_iq=1.00:1.00 mixed_ref=1 me_range=16 chroma_me=1 tcelldct=1 bsdct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=2 threads=1 lockahead_thread=1 sliced_thread=0 nr=0 decimate=1 interlaced=0 blur
VLC compile=0 constrained intra=0 bframes=1 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1 qp=0 weightp=2 keyint=250 keyint_min=20 h264_refresh=0 rc_lookahead=40 rc=crf bitrate=1 crf=23.0 qpom
70.60 qpmin=0 qpmax=63 qpstep=4 ip_ratio=1.40 aq=11.00
Output #0, mp4, to '/usr/src/app/media/tempVideoRepository/2022_10_27_08_48_54.mp4.mp4':
Metadata:
major_brand      : isom
minor_version    : 512
compatible_brands: isomiso2mp4
encoder          : Lavf58.45.100
Stream #0:0(und) : Video: h264 (libx264) (avc1 / 0x31637661), yuv420p, 640x480 [SAR 1:1 DAR 4:3], q=1-1, 20 fps, 10240 tbn, 20 tbc (default)
Metadata:
handler_name     : VideoHandler
encoder          : Lavc58.91.100 libx264
Side data:
cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbr:delay: N/A
frame= 336 fps= 31 q=29.0 Lsize= 1104kB time=00:00:16.65 bitrate= 544.1kbits/s spend=1.52s
video:1101kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.407521%
libx264 @ 0x53f8c2295000 frame 1:0 Avg QP:18.54 avar: 10220
libx264 @ 0x53f8c2295000 frame P:125 Avg QP:21.29 avar: 5049
libx264 @ 0x53f8c2295000 frame B:209 Avg QP:22.08 avar: 2275
libx264 @ 0x53f8c2295000 consecutive B-frames: 15.2% 2.4% 7.5% 72.6%
libx264 @ 0x53f8c2295000 mb I 116..4: 25.3% 73.6% 0.3%
libx264 @ 0x53f8c2295000 mb P 116..4: 2.6% 27.0% 0.1% P16..4: 41.4% 9.1% 6.5% 0.0% 0.0% skip:21.4%
libx264 @ 0x53f8c2295000 mb B 116..4: 1.4% 8.9% 0.0% B16..8: 33.5% 2.6% 0.2% direct: 4.4% skip:48.8% L0:59.0% L1:38.3% BI: 2.6%
libx264 @ 0x53f8c2295000 8x8 transform intra:84.2% inter:96.3%
libx264 @ 0x53f8c2295000 coded y,u,v,w,h intra: 32.4% 60.1% 2.4% inter: 16.1% 32.5% 0.2%
libx264 @ 0x53f8c2295000 i16 v,h,d,c,p: 11% 49% 20% 20%
libx264 @ 0x53f8c2295000 i8 v,h,d,c,d,d,dd,zz,vr,h,d,v1,h: 10% 30% 46% 1% 1% 1% 1% 1%
libx264 @ 0x53f8c2295000 i4 v,h,d,c,d,d,dd,vr,h,d,v1,h: 51% 36% 7% 1% 1% 1% 2% 0%
libx264 @ 0x53f8c2295000 i8c dc,h,v,p: 43% 33% 22% 2%
libx264 @ 0x53f8c2295000 weighted P-frames: 20.0% 80.0%
libx264 @ 0x53f8c2295000 ref P L0: 54.7% 7.4% 25.6% 12.3%
libx264 @ 0x53f8c2295000 ref B L0: 75.4% 10.3% 6.4%
libx264 @ 0x53f8c2295000 ref B L1: 80.5% 0.5%
libx264 @ 0x53f8c2295000 kb/s:536.72
ec2-user@ip-172-31-41-173 SmartHomeCam$

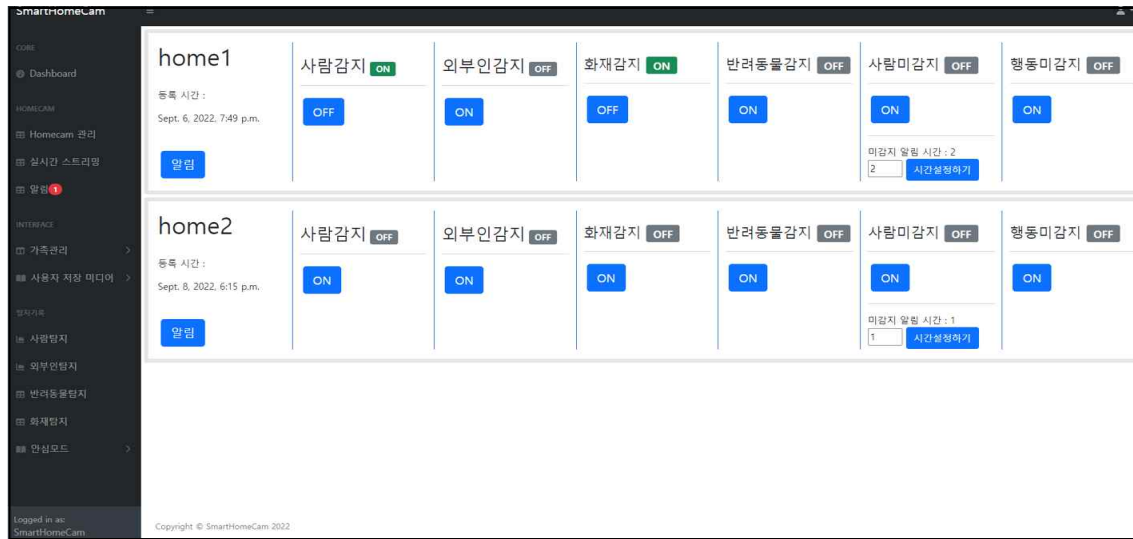
i-034de7d7553f404e2 (SmartHomeCam)
PublicIP: 43.200.29.147 PrivateIP: 172.31.41.173
```

FFmpeg로 코덱 변경

이미지캡처 기능의 경우 이미지를 저장하고 불러오는데 문제가 없었다. 하지만 html5의 video 태그의 표준 비디오 코덱은 MP4 H.264이다. 따라서 서버에서 openCV VideoWriter를 사용해 비디오를 저장할 때 H.264 코덱으로 지정해야 한다. 로컬 환경에서 개발할 때는 기본적으로 윈도우 운영체제 상에서 진행했다. 따라서 H.264 코덱을 이용해 동영상을 저장하는 것이 가능했다. 하지만 리눅스 기반 도커를 사용해 AWS EC2에 배포할 때 리눅스 운영체제에서는 H.264 코덱 사용이 불가능했다. 이 문제를 해결하기 위해서 mp4v 코덱으로 동영상을 저장 후 해당 동영상을 코덱을 FFmpeg로 코덱을 변경 후 AWS S3 Bucket에 업로드 하는 방식으로 해결했다.

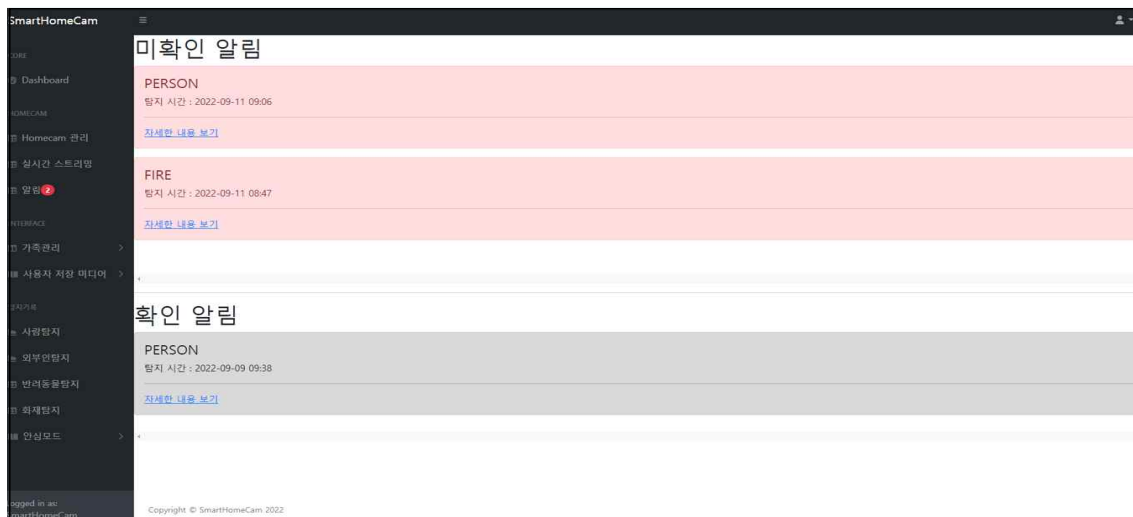
FFmpeg은 디지털 음성 스트림과 영상 스트림에 대해서 다양한 종류의 형태로 기록하고 변환하는 컴퓨터 프로그램이다. FFmpeg은 명령어를 직접 입력하는 방식으로 동작하며 여러 가지 자유 소프트웨어와 오픈소스 라이브러리로 구성되어 있다. 이 프로젝트의 이름은 MPEG 영상 표준화 그룹에서 유래했고, "mpeg" 앞에 붙은 "FF"는 패스트 포워드(fast forward)를 의미한다.

4. 홈카메라 모드



홈카메라 관리 페이지

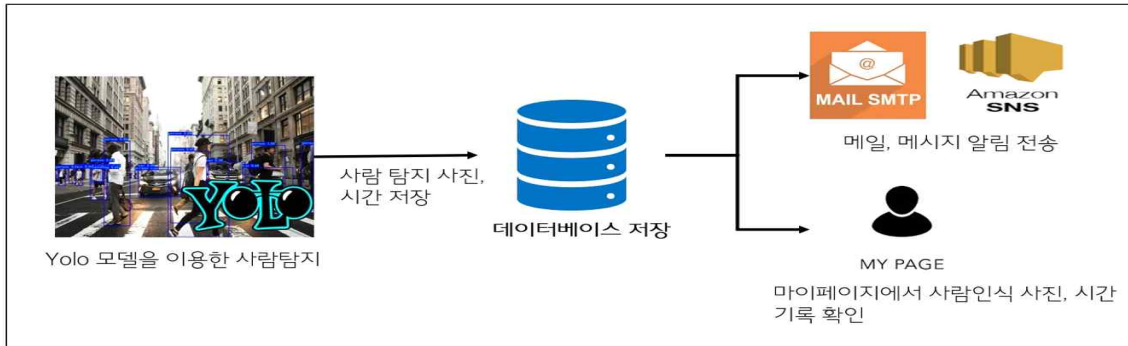
홈카메라 모드는 5가지 모드가 있다. 사람감지/외부인 감지/화재감지/반려동물 감지/안심모드이다. 안심모드의 경우 사람 미감지와 행동 미감지로 구성된다. 홈카메라 관리 페이지에서는 사용자에게 등록된 홈카메라를 관리할 수 있는 페이지로 특정 홈카메라의 알림 확인과 모드를 제어할 수 있다. 버튼을 통해 특정 모드를 on/off 할 수 있고, 사람 미감지 모드는 미감지 기간 일수를 설정할 수 있다.



특정 홈카메라 알림 페이지

홈카메라 관리 페이지에서 특정 홈카메라의 알림 버튼을 클릭하면 해당 홈카메라가 만든 미확인 알림 / 확인 알림을 구분해서 확인할 수 있다.

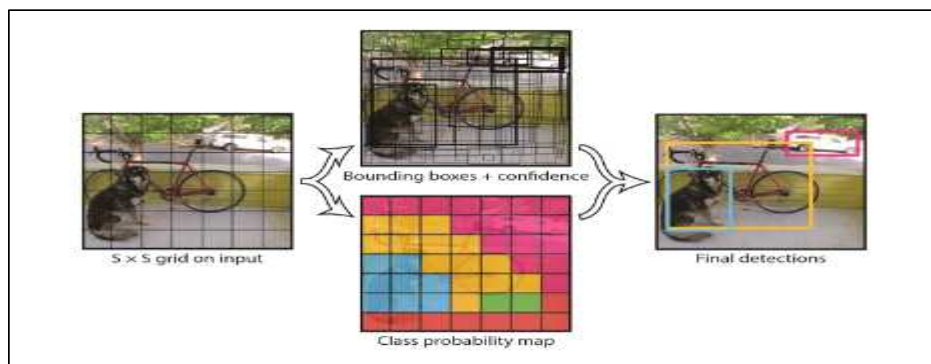
(1). 사람 탐지 모드



사람 탐지 동작

사람 탐지 모드는 YOLO 알고리즘을 기반으로 사람을 탐지해 사람이 탐지되면 탐지 사진과 시간을 데이터베이스에 저장된다. 이때 데이터는 AWS RDS에서 저장되고, 이미지는 AWS S3에 저장된다. 사용자는 이메일과 SMS 메시지로 알림을 받고 마이페이지에서 사진을 확인해 상황을 판단할 수 있다.

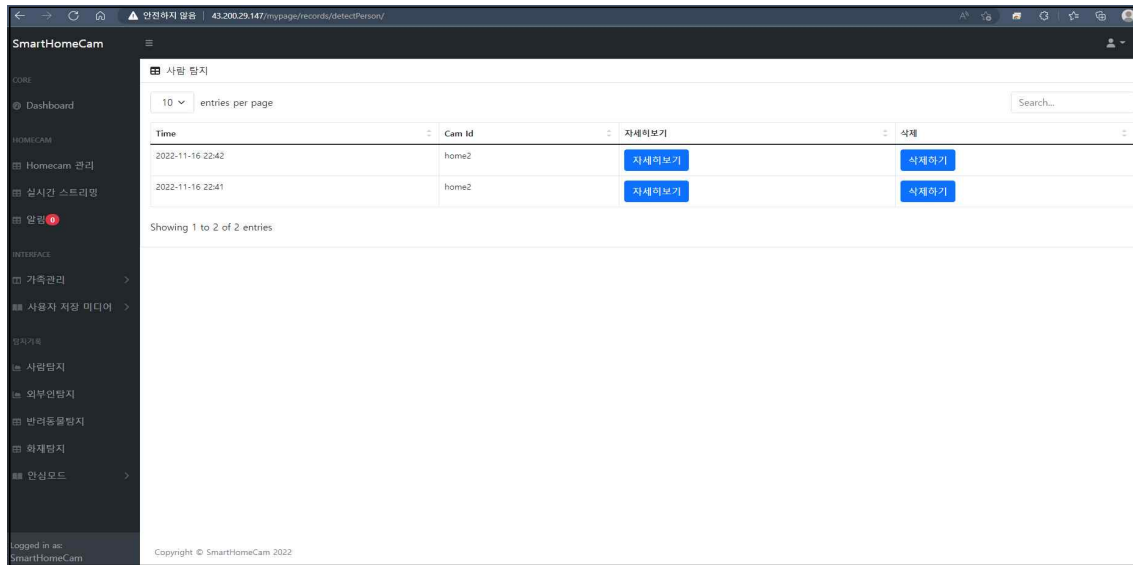
1인 가구의 경우 외출 중에 사람 탐지 모드를 사용하면 외부인 침입 범죄를 빠르게 인지하고 대응할 수 있다. 또 사진이 클라우드에 저장되기 때문에 향후 범죄 수사에 중요한 증거로 사용될 수 있다.



YOLO

이미지에서 객체의 위치를 추정하는 영역 제안(Regional Proposal)과 해당 위치에 있는 객체가 무엇인지 확인하는 분류(classification)을 동시에 수행하는 One-Stage Detector인 YOLO는 빠른 속도와 상대적으로 높은 정확도를 가지고 있어 실시간 객체 탐지에 사용될 수 있다.

YOLO 알고리즘의 과정은 먼저 이미지를 S×S의 정사각형으로 나누고 바운딩 박스, 박스 당 confidence 값, 클래스 분류 확률을 구한다. 여기서 임계값 이상의 박스만 표시한다.



사람 탐지 조회 페이지

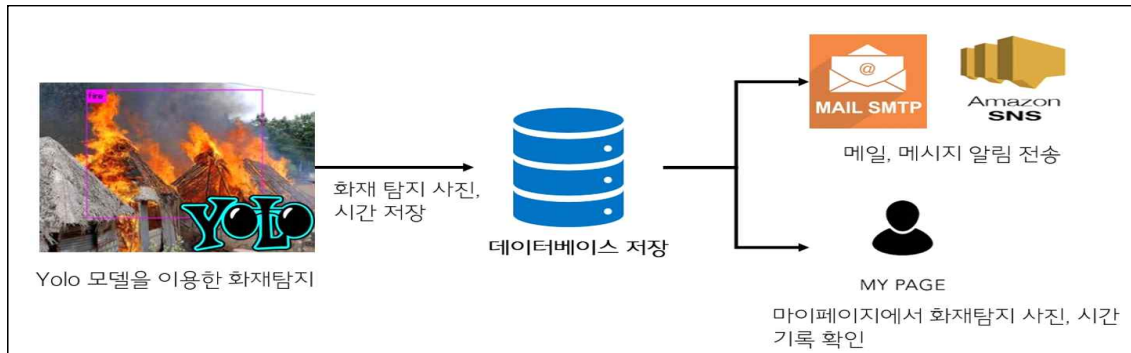
마이페이지 사이드바의 사람탐지를 클릭하면 사람 탐지 내역을 조회할 수 있다. 자세히 보기 버튼을 클릭하면 아래그림 페이지로 이동하고, 삭제하기 버튼을 클릭하면 삭제된다.



사람 탐지 조회 상세페이지

자세히보기 버튼을 클릭하면 위 사진 페이지로 이동된다. 탐지시간, 홈카메라 ID를 확인할 수 있다. 원본사진과 함께 사람 객체를 바운딩한 탐지 객체 바운딩 사진을 함께 볼 수 있다. 삭제하기 버튼을 클릭하면 삭제된다.

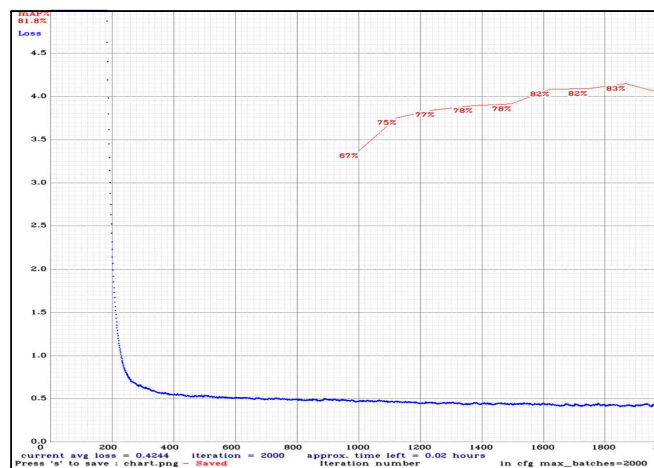
(2). 화재 탐지 모드



화재 탐지 동작

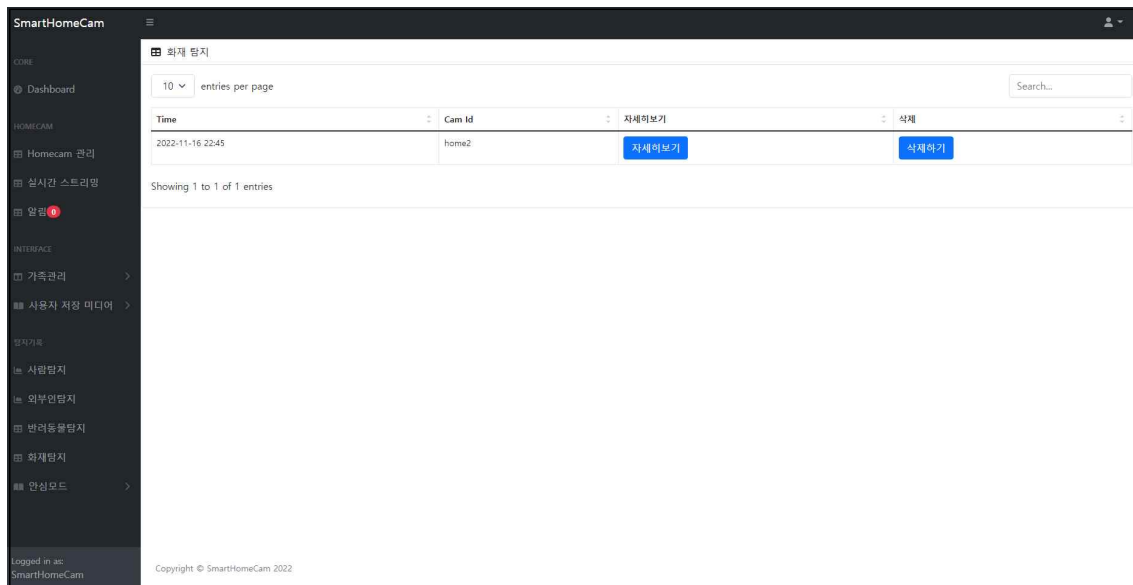
화재 탐지 모드는 YOLO 알고리즘을 기반으로 불꽃을 탐지 후 탐지 사진과 시간을 데이터베이스에 저장된다. 이때 데이터는 AWS RDS에서 저장되고, 이미지는 AWS S3에 저장된다. 사용자는 이메일과 SMS 메시지로 알림을 받고 마이페이지에서 사진을 확인해 상황을 판단할 수 있다.

1인 가구의 경우 외출 중에 기존의 열 탐지 기반의 탐지에서 벗어나, 이미지를 기반으로 화재를 탐지해 작은 불꽃을 화재 발생으로 인식할 수 있기 때문에 초기 화재도 탐지가 가능해 빠른 대응으로 인명/재산 피해를 최소화할 수 있다.



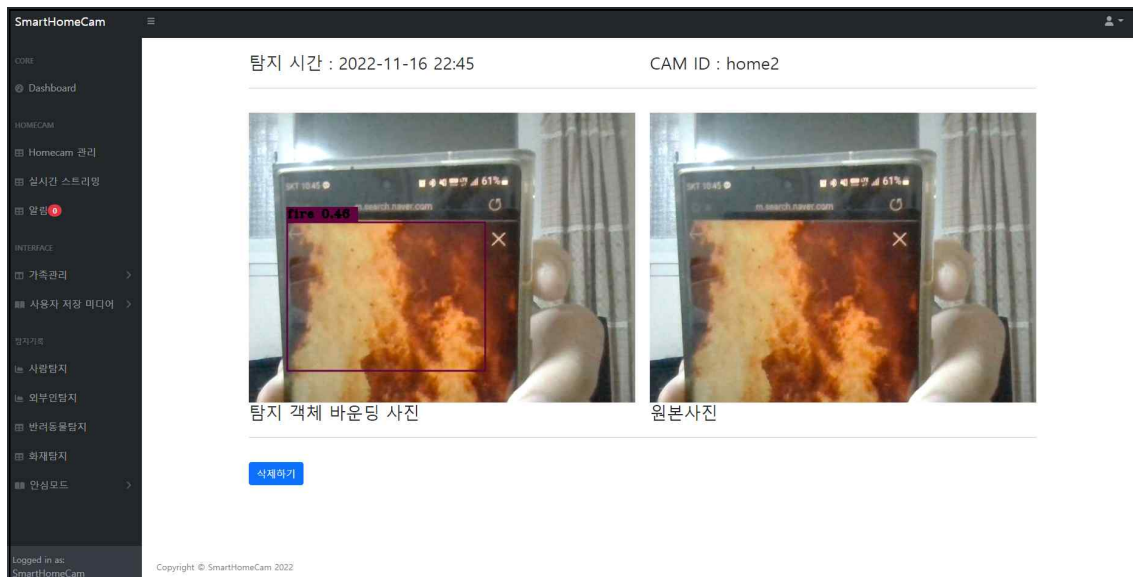
YOLO v4 tiny Fire custom train

오픈소스로 개발된 YOLO의 기본모델에는 화재 객체를 탐지할 수 없기 때문에 custom 훈련을 해야한다. Kaggle에서 얻은 화재 불꽃 이미지 학습 데이터를 세가지 방법으로 증강(augmentation)하여 학습 정밀도를 높였다. 밝기 조절, 가우시안 필터의 커널 크기 조절을 통한 블러링, 불꽃 사진 합성을 통해 화재 상황 데이터를 추가로 생성하고, 이 데이터를 기반으로 화재 검출 모델을 학습하였다.



화재 탐지 조회 페이지

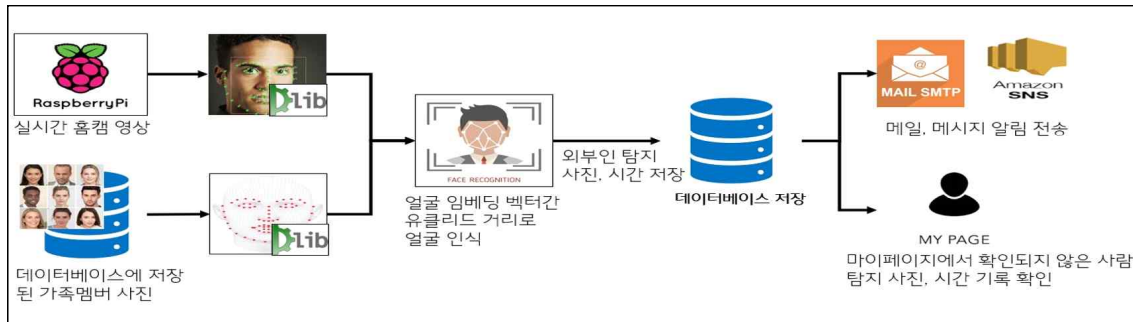
마이페이지 사이드바의 화재탐지를 클릭하면 화재 탐지 내역을 조회할 수 있다. 자세히 보기 버튼을 클릭하면 아래그림 페이지로 이동하고, 삭제하기 버튼을 클릭하면 삭제된다.



화재 탐지 조회 페이지

자세히보기 버튼을 클릭하면 위 사진 페이지로 이동된다. 탐지시간, 홈카메라 ID를 확인할 수 있다. 원본사진과 함께 화재(불꽃) 객체를 바운딩한 탐지 객체 바운딩 사진을 함께 볼 수 있다. 삭제하기 버튼을 클릭하면 삭제된다.

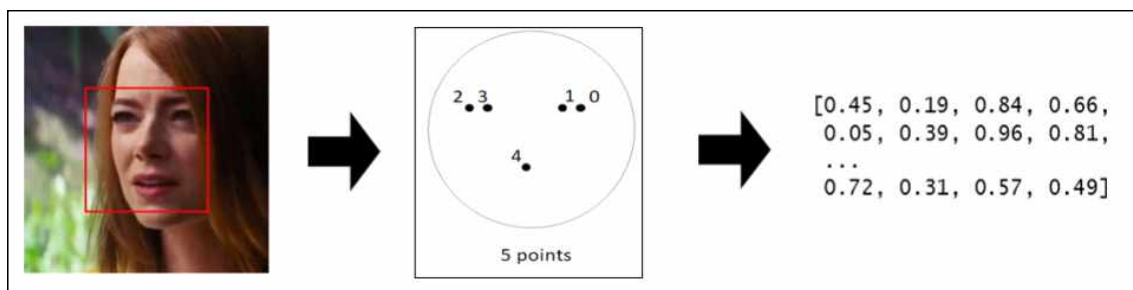
(3). 외부인 탐지 모드



외부인 탐지 동작

외부인 탐지 모드는 가족 구성원의 얼굴과 실시간으로 탐지된 얼굴을 비교해 외부인이라 판단되면 탐지 사진과 시간을 데이터베이스에 저장된다. 이때 데이터는 AWS RDS에서 저장되고, 이미지는 AWS S3에 저장된다. 사용자는 이메일과 SMS 메시지로 알림을 받고 마이페이지에서 사진을 확인해 상황을 판단할 수 있다.

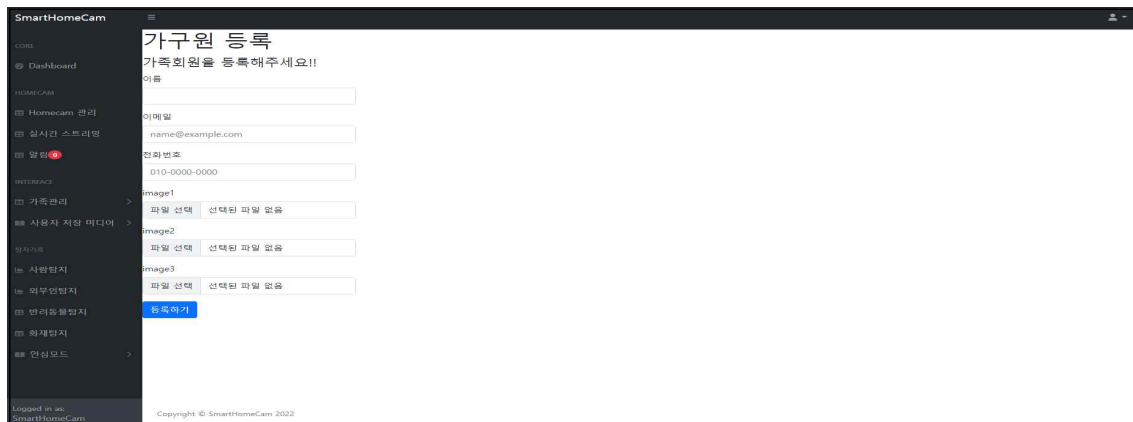
2명 이상 거주중인 가구는 가족 구성원의 얼굴 사진을 등록함으로써 맞춤형 보안 솔루션을 제공할 수 있다.



임베딩 기반의 얼굴인식

hog 기반의 dlib face detector를 사용해 얼굴을 인식하고 dlib을 이용한 facial landmark detection(5-point facial landmark)을 이용해 왼쪽 눈 2개, 오른쪽 눈 2개, 코 1개 점을 DNN을 이용해 얼굴을 128개 벡터로 수치화한다. 데이터베이스에 저장된 가족 구성원 얼굴 벡터와 실시간으로 인식된 얼굴 벡터의 유클리디안 거리가 0.5 미만일 경우 외부인으로 판단한다.

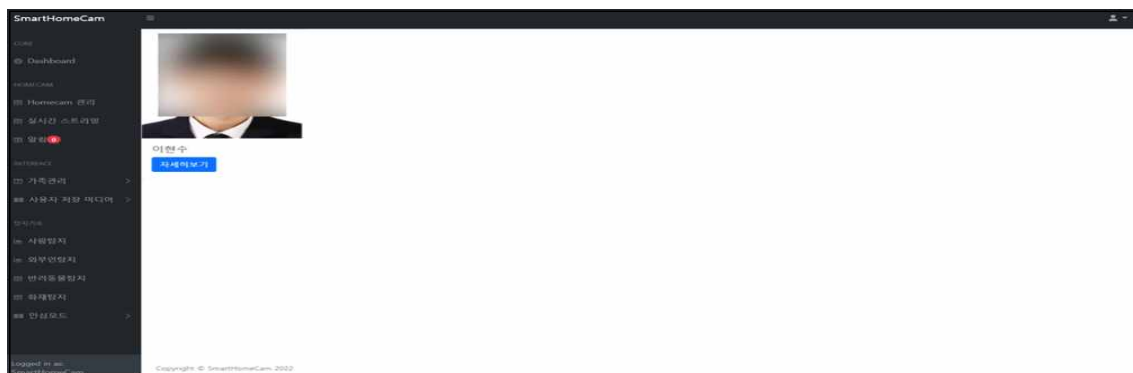
128차원의 벡터를 numpy로 연산하기 위해서 numpy array로 변환해준 후 각 벡터간의 거리를 L2 Norm Distance(유클리디안 거리)로 계산한다. L2 Norm Distance는 각 차원의 차이를 제곱한 뒤 합한 후 제곱근을 구한 값으로 2차원 유클리디안 거리를 다차원으로 확장시킨 것이다.



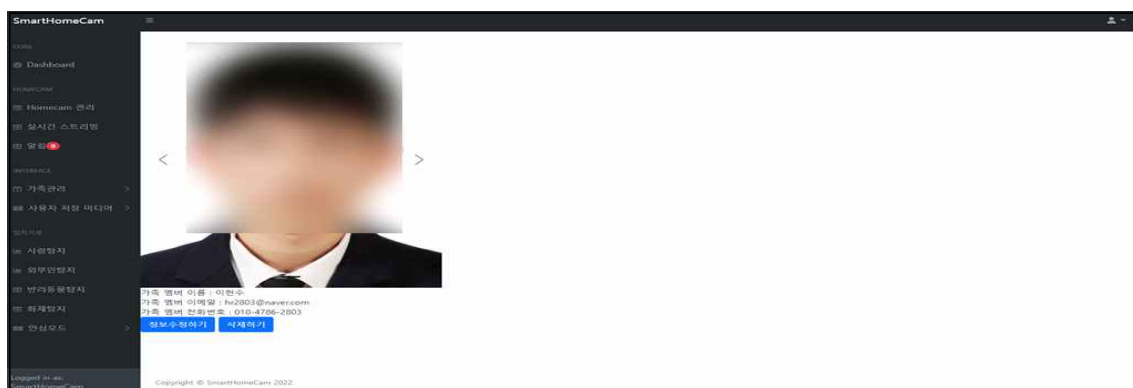
The image shows the '가구원 등록' (Member Registration) page of the SmartHomeCam system. The page has a dark sidebar on the left with a menu including 'HOME', 'Dashboard', 'SmartCam', 'Homecam 관리', '실시간 스트리밍', '알림', 'INTERACE', '가족관리', '사용자 저장 미디어', '설정', '사람탐지', '외부연동장치', '반려동물탐지', '화재탐지', and '단일 모드'. The main content area is titled '가구원 등록' and '가족회원을 등록해주세요!!'. It contains a form with the following fields: '이름' (Name), '이메일' (Email, pre-filled with 'name@example.com'), '전화번호' (Phone Number, pre-filled with '010-0000-0000'), and three image upload sections labeled 'image1', 'image2', and 'image3'. Each image section has a '파일 선택' (Select File) button and a status indicator '선택된 파일 없음' (No file selected). At the bottom of the form is a blue '등록하기' (Register) button. The footer shows 'Logged in as: SmartHomeCam' and 'Copyright © SmartHomeCam 2022'.

가구원 등록 페이지

가구원 등록 페이지에서 이름, 이메일, 전화번호, 사진을 등록할 수 있다. 사진은 최소 한 장부터 최대 세장까지 등록이 가능하다. 여기에 등록한 이메일과 전화번호를 기반으로 알림이 전송되고, 얼굴 사진을 기반으로 외부인을 판별한다.

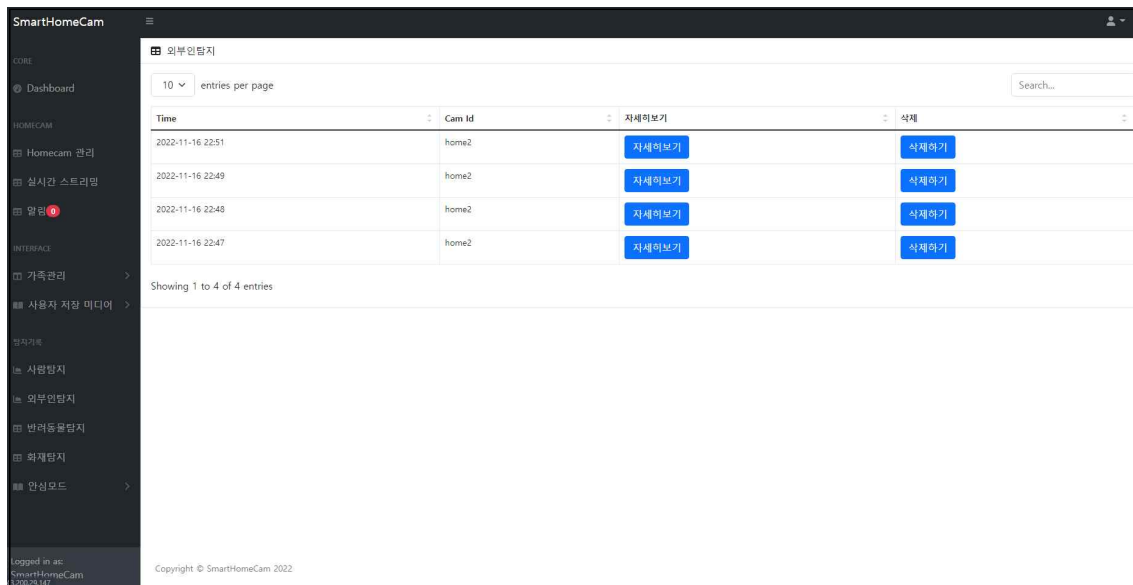


등록된 가구원 조회 페이지



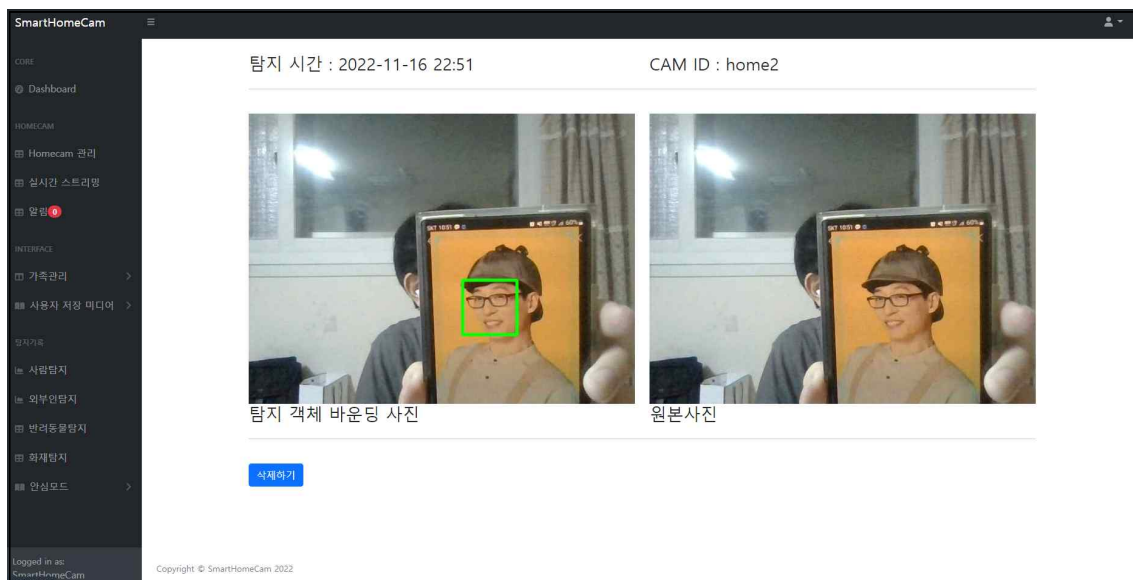
등록된 가구원 조회 상세 페이지

등록한 가구원을 조회할 수 있다. 정보 수정하기 버튼을 눌러 가구원 정보를 수정할 수 있고, 삭제하기 버튼을 누르면 해당 가구원의 정보는 삭제된다.



외부인 탐지 기록 페이지

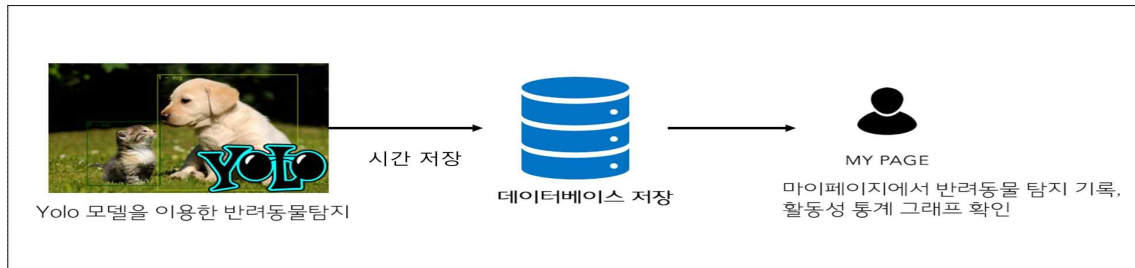
마이페이지 사이드바의 외부인탐지를 클릭하면 외부인 탐지 내역을 조회할 수 있다. 자세히 보기 버튼을 클릭하면 아래그림 페이지로 이동하고, 삭제하기 버튼을 클릭하면 삭제된다.



외부인 탐지 상세 조회 페이지

자세히보기 버튼을 클릭하면 위 사진 페이지로 이동된다. 탐지시간, 홈카메라 ID를 확인할 수 있다. 원본사진과 함께 외부인 얼굴 바운딩 사진을 함께 볼 수 있다. 삭제하기 버튼을 클릭하면 삭제된다.

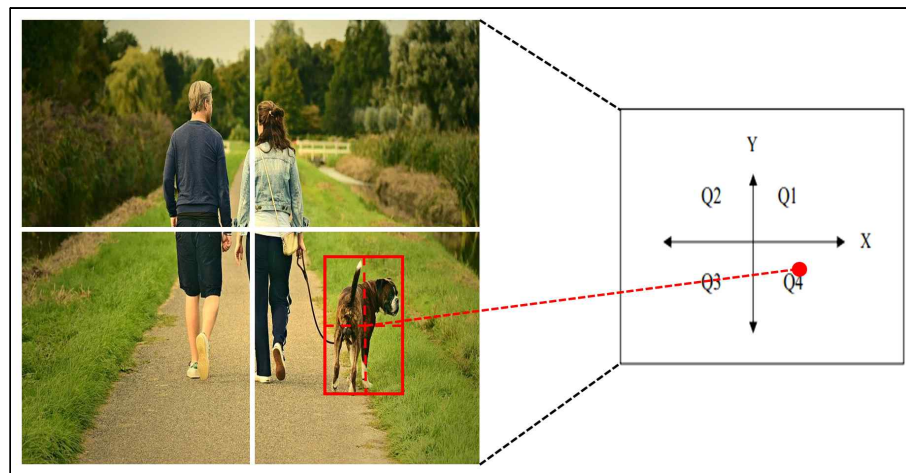
(4). 반려동물 탐지 모드



반려동물 탐지 동작

반려동물 탐지 모드는 YOLO 알고리즘을 기반으로 반려동물을 탐지해 활동성을 확인할 수 있다. 이때 반려동물을 탐지할 때 개인 맞춤형 반려동물을 대상으로 탐지하는 것이 아닌 YOLO 기본 모델에 학습된 강아지, 고양이를 탐지해 데이터베이스에 저장한다.

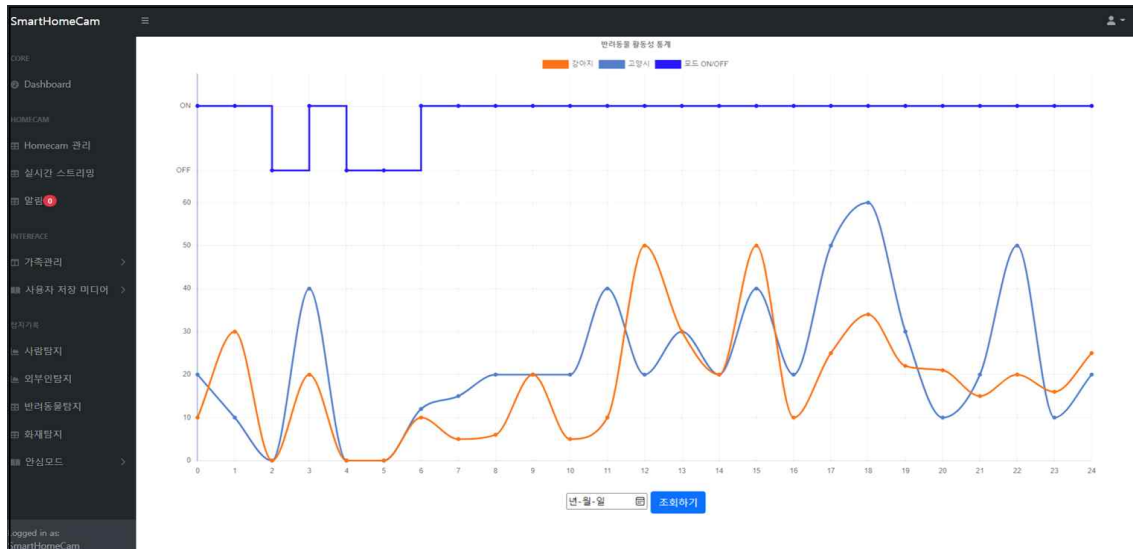
이를 통해 집에 혼자 방치되는 반려동물의 활동성을 확인해 전반적인 정서적/육체적 건강상태를 유추할 수 있다.



이미지에서 반려동물 위치 계산

반려동물의 활동성을 계산하기 위해서 반려동물이 탐지되면 반려동물 객체의 바운딩 박스의 중앙점을 이용해 4사분면을 판단해 사분면 정보를 데이터베이스에 저장한다.

반려동물의 활동성 통계 그래프를 그릴 때 다음과 같은 알고리즘을 구현했다. 우선 반려동물이 카메라에 많이 포착되었더라도 같은 지점에서 잠을 자고 있는 상황일 수도 있기 때문에 같은 시간대에 탐지된 location의 연속점이 전체 수의 50%일 경우 해당 시간에는 잠을 잔다고 판단한다. 그게 아니라면 location(1, 2, 3, 4)의 퍼센트를 구해 %가 골고루 퍼질 경우 높은 활동성으로 판단하고 편향될수록 활동성이 낮다고 판단한다.



반려동물 활동성 통계 그래프

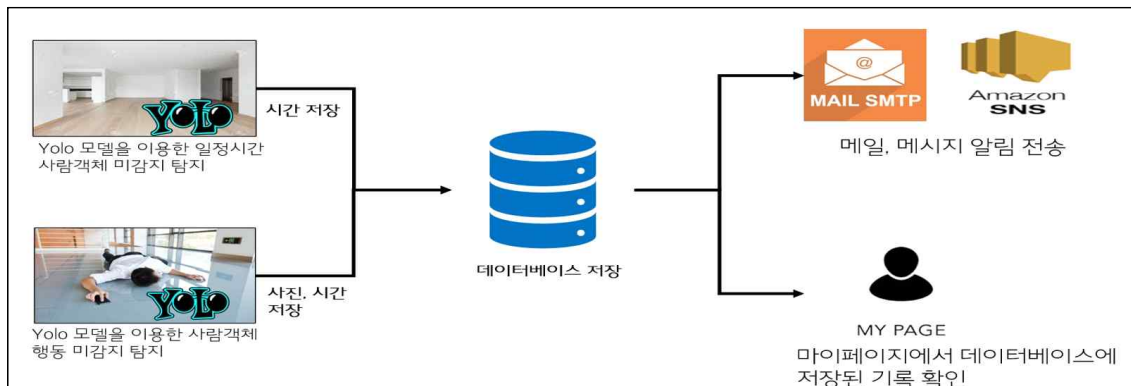
마이페이지에서 사이드바의 반려동물탐지 탭을 클릭하면 반려동물 활동성 통계 그래프를 볼 수 있다. 원하는 날짜를 선택해 조회하기 버튼을 클릭하면 활동성 통계를 볼 수 있다. 그래프는 강아지, 고양이로 구분되고 모드를 OFF한 경우와 움직이지 않은 경우의 혼돈을 방지하기 위해 반려동물 탐지모드 ON/OFF 기록도 그래프에 표시된다. 데이터를 요청할 때 Ajax 통신을 이용해 요청하기 때문에 페이지 새로고침 없이 그래프가 그려진다. 이때 chart.js를 사용해 그래프를 구현했다.

장소	시간	camid
강아지	2022-05-23 10:09	home1
강아지	2022-05-21 09:20	home1
강아지	2022-05-21 09:20	home1
강아지	2022-05-21 09:20	home1
강아지	2022-05-21 09:20	home1
강아지	2022-05-21 09:20	home1
강아지	2022-05-21 09:19	home1
강아지	2022-05-21 09:03	home1
강아지	2022-05-21 09:03	home1
강아지	2022-05-21 09:03	home1

반려동물 탐지 기록 확인 (표)

활동성 그래프 아래에는 반려동물 탐지 기록이 표 형태로 보여진다. 어떤 홈카메라에서 무슨 종(강아지/고양이)을 언제 탐지했는지 출력된다.

(5). 안심 모드



안심모드 동작

안심모드는 두가지 기능이 있다. 일정 시간 사람 미감지와 행동 미감지 탐지이다. 이를 통해 혼자 사는 부모님 / 노인 / 장애인 / 보호 대상에 대해 사고와 고독사를 일정 부분 빠르게 인지할 수 있다.

시간	미탐지 시간	홈 ID
2022-06-06 19:50	30	home1
2022-06-06 19:39	30	home1
2022-06-06 19:37	30	home1
2022-06-06 19:32	30	home1
2022-06-06 19:31	30	home1

Showing 1 to 5 of 5 entries

일정시간 사람 미감지 기록 조회 페이지

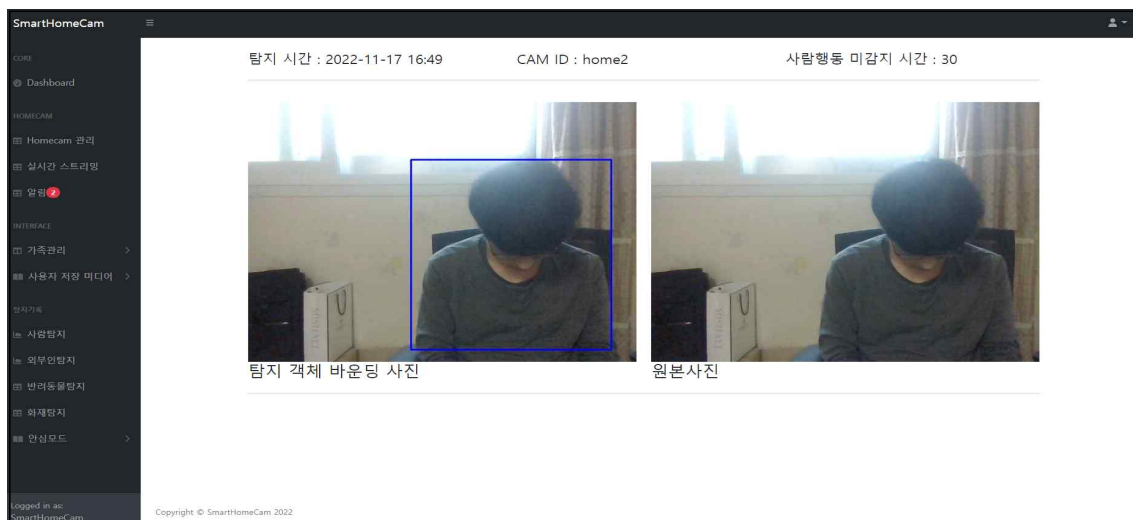
마이페이지에서 사이드바의 안심모드 탭에서 사람 미감지 탭을 클릭하면 일정 시간 사람 미감지 기록 확인 페이지로 이동한다. 이 모드의 경우 이미지 없이 시간, 미탐지 시간, 탐지한 홈카메라 ID를 표로 출력된다. 위 사진에서는 테스트를 위해 30초로 설정해 테스트 했다. 하지만 실제 구현은 1일단위로 최소 1일부터 최대 7일까지 시간을 설정할 수 있다.

Time	Cam id	행동 미감지 시간	자세히보기	삭제
2022-11-17 16:49	home2	30	자세히보기	삭제하기
2022-11-17 11:21	home2	30	자세히보기	삭제하기
2022-06-13 09:41	home1	30	자세히보기	삭제하기
2022-06-13 09:36	home1	30	자세히보기	삭제하기

Showing 1 to 4 of 4 entries

안심모드 사람 행동 미감지 조회 페이지

사람 행동 미감지의 경우 일정 시간(8시간) 카메라에 인식된 사람 객체의 움직임이 없는 상황을 탐지한다. 마이페이지 사이드바의 안심모드 탭에서 행동 미감지 탭을 클릭하면 사람 행동 미감지 탐지 기록을 조회할 수 있습니다. 자세히 보기 버튼을 클릭하면 아래그림 페이지로 이동하고, 삭제하기 버튼을 클릭하면 삭제된다.

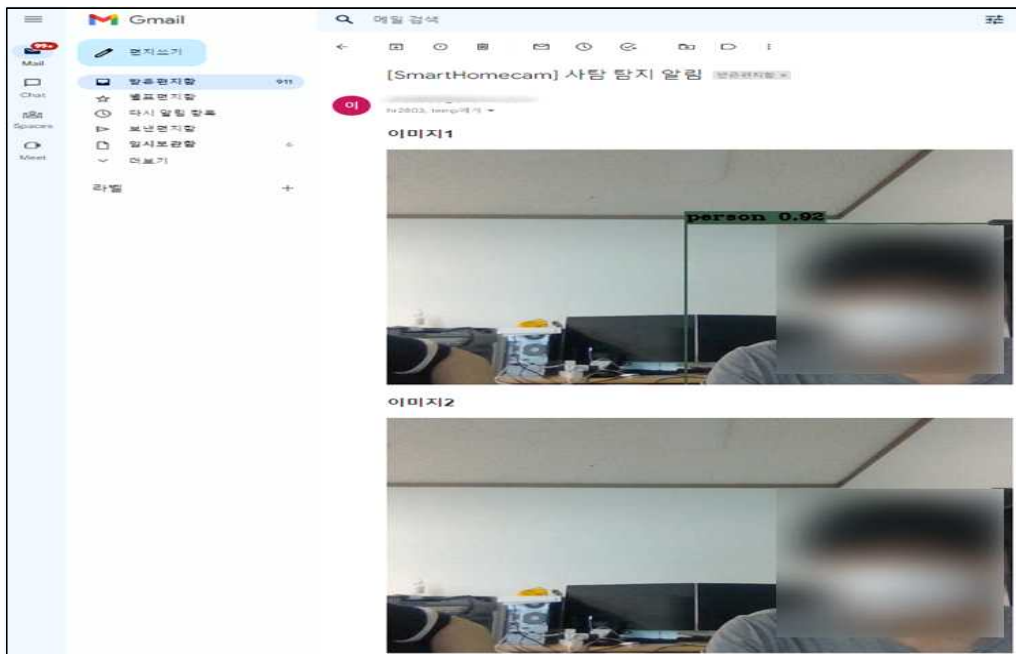


안심모드 사람 행동 미감지 조회 상세 페이지

움직임이 없는 사람 객체를 바운딩한 사진과 원본 사진을 확인할 수 있다. 30초 동안 사람 행동이 없는 경우를 가정해 테스트를 진행했다. 사람 행동 미감지 기능은 기절, 실신, 넘어짐, 고독사 등의 상황을 빠르게 파악해 빠른 상황 인식과 대처를 가능하게 해준다.

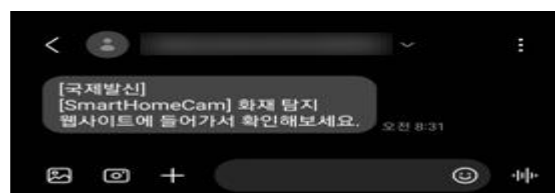
5. 이메일 · SMS 알림

홈카메라 모드의 특정 상황 발생 시 가족 구성원 등록 때 함께 저장한 이메일, 전화번호로 이메일, SMS 알림이 전송된다. 사람/외부인/화재/일정시간 사람 미감지/사람 행동 미감지의 경우 이메일 / SMS 알림이 전송된다.



사람 탐지 이메일 알림

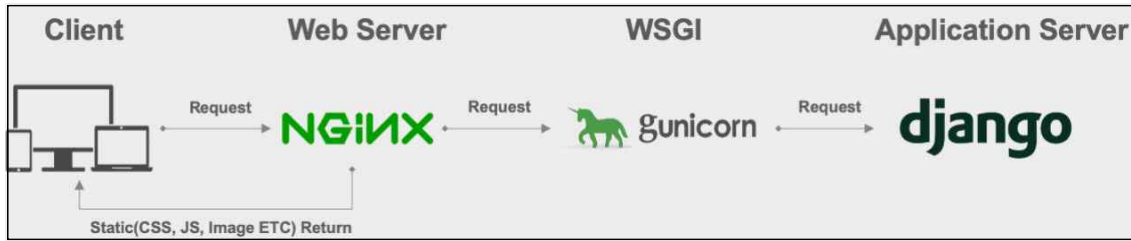
일정 시간 사람 미감지는 텍스트 내용으로만 구성되고, 그 외 나머지는 이미지가 함께 전송된다. 이메일 알림은 구글 Gmail SMTP(Simple Mail Transfer Protocol) 서버를 사용해 전송한다. 파이썬에 기본 내장된 smtplib 모듈을 사용해 구현했다.



화재 탐지 SMS 알림

SMS 알림은 모두 텍스트로만 구성되고, AWS SNS의 SMS 전송 기능은 아시아 태평양(서울) 리전(ap-northeast-2)에서 사용할 수 없어, 아시아 태평양(도쿄) 리전(ap-northeast-1)을 이용하기 때문에 국제발신으로 전송된다.

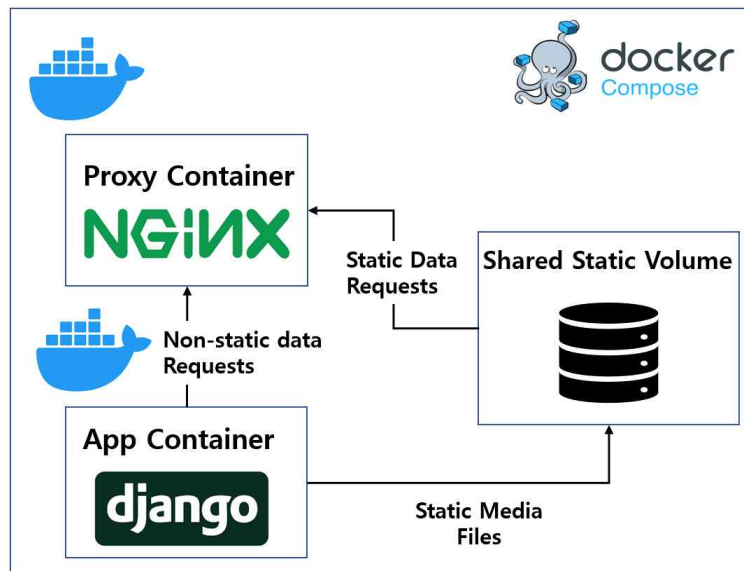
6. AWS 클라우드 배포



웹서버 요청 흐름도

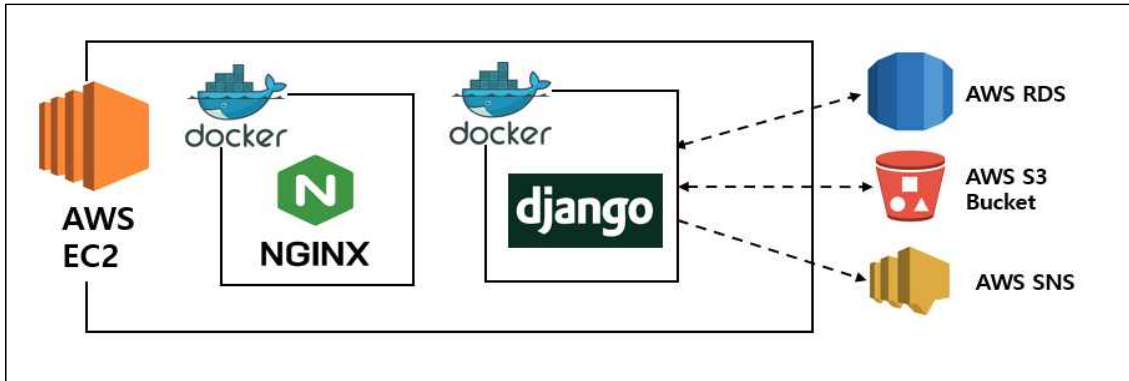
Django에서 제공하는 runserver는 개발용으로 사용하는 경량 웹 서버로 실제 production 환경에서는 적합하지 않다. 따라서 Nginx 웹서버를 사용하고 그 웹서버와 Django 어플리케이션을 연결해 주는 Gunicorn WSGI를 이용해 배포했다.

Gunicorn과 같은 WSGI만 있어도 HTTP Request를 처리할 수 있지만, Web Server에서 제공하는 Static, Dynamic Request에 대한 처리와 적은 메모리 사용이라는 장점을 이용할 수 없으므로, 실제 production 환경에서는 Web Server를 함께 사용하는 것이 좋다.



docker 구성도

Django 웹서버를 실행하기 위해서는 opencv, dlib 라이브러리 등이 필요하다. 따라서 환경표준화를 고려해 도커를 사용해 배포하기로 했다. 하나의 Container안에 Django 어플리케이션과 Nginx를 함께 실행할 수도 있지만, 각각의 Container로 분리하고 따로 관리하는 것이 효율적이다. Docker-compose를 이용해 Django와 Nginx Container를 생성하고 이들을 연결해주었다.



AWS Service 구성도

클라우드 환경에 배포하기 위해 AWS EC2, AWS RDS, AWS S3 Bucket, AWS SNS를 사용했다.

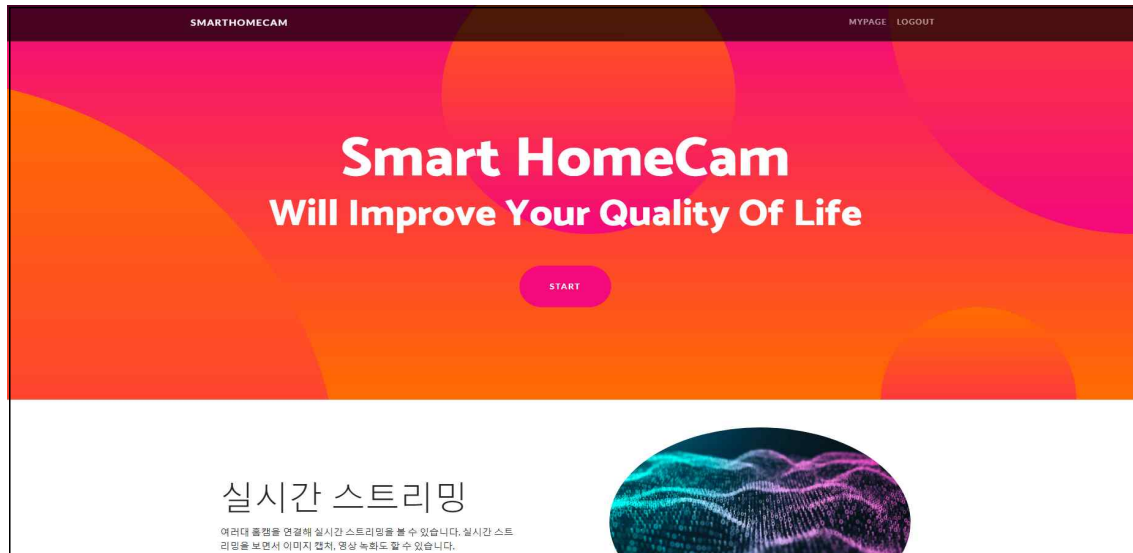
Amazon Elastic Compute Cloud(Amazon EC2)는 크기 조정이 가능한 가상 머신으로 IaaS(서비스형 인프라)에 속한다. Github Actions에서 프로젝트를 도커로 빌드해 컨테이너화 하고 도커허브에 업로드한다. 그 후 AWS EC2에 접속해 도커허브에 올라온 도커 이미지를 다운받아 실행한다.

Amazon RDS는 클라우드에서 관계형 데이터베이스를 구성하고 운영하는 관리형 서비스이다. 비용 효율적이고 크기 조정 가능한 용량을 제공하는 한편, 시간이 많이 소모되는 관리 작업을 자동화할 수 있다. Amazon RDS는 MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, Oracle의 6개 데이터베이스를 지원한다. 본 프로젝트에서는 MariaDB를 사용했다. MariaDB는 MySQL 기반으로 만들어졌기 때문에 쿼리를 비롯한 전반적인 사용법은 MySQL과 유사하다. MariaDB는 MySQL보다 동일 하드웨어 사양으로 보다 향상된 성능을 발휘하고 다양한 기능과 스토리지 엔진이 있기 때문에 MariaDB를 선택했다.

Amazon S3는 객체 수준 스토리지로 데이터를 버킷이라는 리소스 내에 객체로 저장한다. 영구 스토리지로 URL(Uniform Resource Locator)을 통해 사용할 수 있으며 어디서든 액세스할 수 있다. 본 프로젝트에서 발생하는 모든 이미지, 동영상은 버킷에 저장해 URL을 통해 접근할 수 있도록 했다.

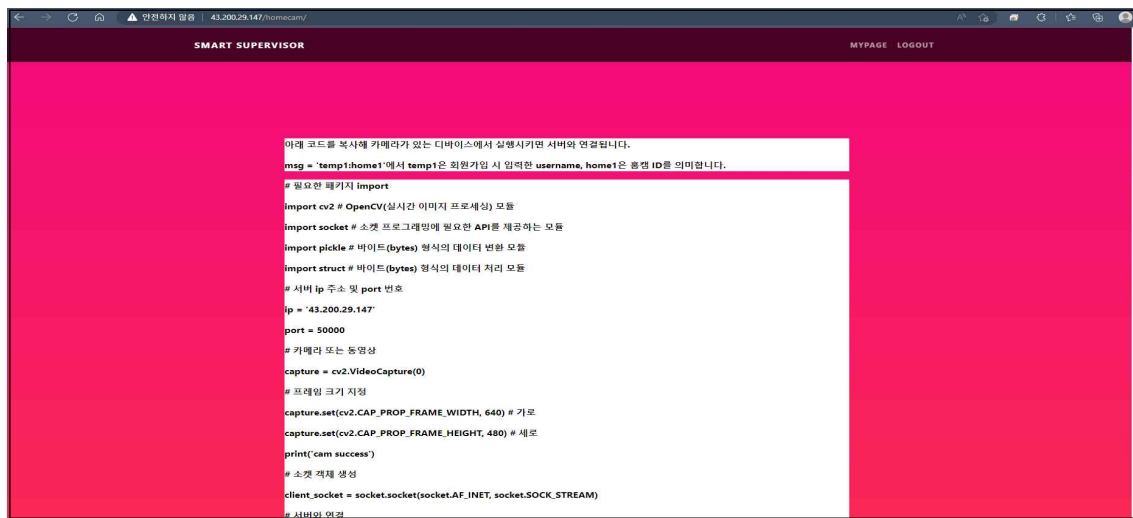
Amazon Simple Notification Service(SNS)는 게시자(생상자)에서 구독자(소비자)로 메시지를 전송하는 관리형 서비스이다. 클라이언트는 SNS 주제를 구독하고 Amazon Kinesis Data Firehose, Amazon SQS, AWS Lambda, HTTP, 이메일, 모바일 푸시 알림 및 모바일 문자 메시지(SMS)와 같이 지원되는 엔드포인트 유형을 사용하여 게시된 메시지를 수신할 수 있다. 본 프로젝트에서는 문자 인증을 통해 구독한 서비스 사용자에게 문자 메시지(SMS) 알림을 전송한다.

IV. 개발 결과



Smarthomecam 사이트 메인페이지

AWS를 사용해 클라우드에 배포했다. <http://43.200.29.147>에 접속해서 홈카메라 서비스를 이용할 수 있다. 단, 비용문제로 서버를 중지한 경우 웹사이트 접속이 불가능할 수도 있다.



임베디드 보드에서 실행하는 소스코드

본 프로젝트의 기능을 사용할 때 임베디드 보드가 필수적으로 필요하지는 않다. 파이썬 파일이 구동될 수 있는 환경과 카메라만 있으면 된다. 메인페이지에서 START 버튼을 클릭해서 나오는 소스코드를 카메라가 부착된 컴퓨터에서 실행하면 서비스 이용이 가능하다.

V. 결론

Django 프레임워크와 AWS 서비스를 사용해 성공적으로 개발을 완료했다. 로컬 개발 환경과 클라우드 배포 환경의 차이 때문에 발생하는 문제점이 있었다. 따라서 로컬 개발 환경부터 도커 등의 환경으로 일치시켜 개발하는 것이 좋다고 생각한다. 본 프로젝트는 가정용 홈카메라이지만 인공지능 알고리즘을 추가하면 산업 현장, 일상적인 공간에서 근로자 안전 모니터링, 사고 / 범죄 탐지 등에 응용이 가능하다. 따라서 본 프로젝트는 범용성과 확장성이 높다.