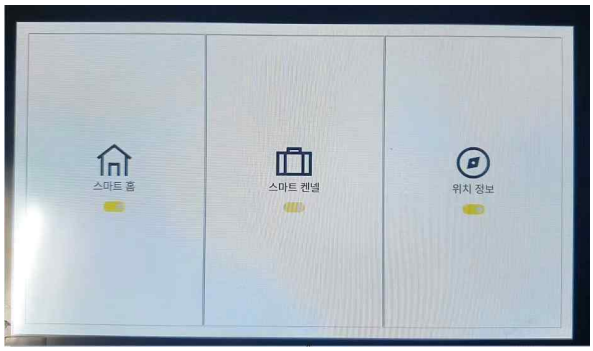
제18회 임베디드SW경진대회 개발완료보고서

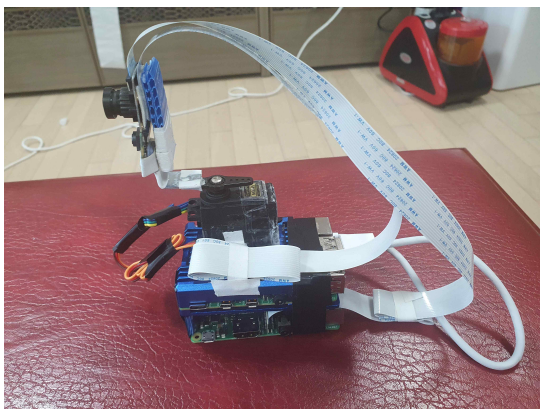
[webOS 기반 차량용 인포테인먼트 솔루션]

□ 개발 요약

팀 명	피쉬앤칩스



차량용 인포테인먼트 기기



스마트 홈 기기

스마트 켄넬 기기

작품명	방울이가 타고있어요
작품설명 (요약)	반려동물을 가정에 두고 차량을 이용하거나, 반려동물과 함께 차량 이동을 할 때 발생할 수 있는 불편함을 해결 할 수 있는 인포테인먼트 기기, 스마트 켄넬, 스마트 홈 기기의 개발
소스코드	https://github.com/Galocg/2020ESWContest_webOS_4016.git
시연동영상	https://youtu.be/oWH7sbvk6Cw

□ 개발 개요

○ **개발 배경 및 동기**

- 현재 반려동물을 기르는 가구의 수가 500만 가구에 육박하고, 이에 따라 반려동물을 키우는 가정의 불편사항도 다양해지고 있습니다. 이 작품은 그 중 반려동물을 기르는 사람이 **반려동물을 가정에 두고 차량을 이용하거나, 반려동물과 함께 차량 이동을 할 때 발생할 수 있는 문제에 초점**을 맞추어 개발하였습니다.

- 저희가 착안한 불편한 포인트 3가지는 다음과 같습니다.

  1) 차량 탑승 시 기존 켄넬은 반려동물이 차량에 탑승했을 때 발생할 수 있는 불안, 구토, 멀미 등의 증세를 확인하기 어려운 단점을 가지고 있다.

  2) 반려동물을 차량 또는 집에 혼자 두어야하는 상황일 때, 보호자는 반려동물의 상태를 확인하기 어려워 불안감을 느낍니다.

  3) 운전 중, 응급상황이나 즉석해서 애완동물 출입 시설에 갈 필요가 있을 경우, 근처 시설을 즉각 찾아내고 길안내를 받는데 어려움이 있습니다.

○ **개발 작품 개요**

- 차량 탑승 시, 제공된 webOS 기기를 스마트 홈 기기, 스마트 켄넬와 연동해 반려 동물의 확인 및 케어 시스템을 제공합니다.

- 차량 탑승 시, 제공된 webOS 기기를 통해 현 위치에서 가까운 애견 편의시설을 제공하고, 길 안내 서비스를 제공합니다.

○ **개발 목표**

- 차량 내부의 인포테인먼트 기기를 통해, 차량에서 최소한의 조작을 통해 펫 케어 서비스를 이용하는 것을 목표로 합니다.

- 스마트 켄넬 기기의 경우, 운전자가 뒷좌석의 켄넬 안 강아지의 상태를 확인하기 힘들 것이며, 강아지가 운전에 의한 스트레스로 인해 멀미, 구토 및 불안 증상을 보일 가능성이 높은 점에 착안하여 개발했습니다.

- 스마트 홈 기기의 경우, 원격으로 집에 있는 강아지와 대화할 수 있으며, 안전을 위해 운전자는 강아지 케어를 위해 최소한의 조작만을 통해 기능을 수행 할 수 있도록 하는 것을 목표로 합니다.

- 차량용 인포테이먼트 장치의 '위치 정보' 기능의 경우 운전자의 현 위치를 기반으로 운전 중에 주변 애견 출입 장소를 검색하기 어려움을 겪는다는 점에 착안해, 근처의 공원, 카페 등 원하는 애견 출입 장소를 알려주고, 안내하며 동시에 최대한 많은 정보를 제공해 줄 수 있도록 합니다.

□ 개발 환경 설명

○ Hardware 구성

- 차량용 인포테인먼트 기기 : Raspberry Pi 4B, 마이크, 터치모니터
- 스마트 홈 기기 : Raspberry Pi 4B, Raspberry Pi 3B+, 마이크, 스피커, Coral USB Accelerator(Google Edge TPU), 라즈베리파이 카메라 2대
- 스마트 켄넬 기기 : Raspberry Pi 3B+, 마이크, 스피커, 라즈베리파이 카메라, DHT 센서

○ Software 구성

- 차량용 인포테인먼트 기기 :

차량용 인포테인먼트 기기는 크게 Views, Components, Actions 세 파트로 나누어집니다. 또한 API에서 서버와 HTTPS 통신을 하기 위해 Axios 라이브러리를 선언해두었습니다.

1) Views : 한 화면을 구성하는 단위로, 다양한 Components의 조합으로 구성됩니다 (MainView,HomeView, KennelView, LocationView)

2) Components : 버튼, 박스 같은 구성 요소들을 의미합니다. 버튼의 디자인, 기능을 구현하고 상태를 유지하지 않는 동작을 수행합니다. (Button, BoxHeader, Video, ...)

3) Actions : 상태를 유지하는 동작을 수행하기 위한 부분입니다. 상태 저장소인 store, 상태를 변경하는 동작을 정의하는 actions, action에 따라 상태를 바꿔주는 reducers로 구성됩니다. Action은 Component와 connect를 이용하여 연결할 수 있습니다. (connectAction, locationAction, webosAction, videoAction)

- 메인 서버 :

1) Node.js express 프레임워크를 통해 클라이언트의 http 요청 처리합니다.
2) RESTful API를 제공합니다
3) SSL 인증서 적용을 통한 https 사용합니다.
4) 서버 로그 기록 모듈 Winstion을 적용합니다.

- DogVomitML 서버 :

DogVomitML 서버는 개가 멀미를 하는지 확인하는 머신러닝 모듈을 내장한 서버입니다. 멀미하는 사진(입을 벌리고 있는 사진, 귀를 젖히고 있는 사진 등)과 일반적인 사진의 두 분류의 강아지 사진 데이터 세트를 학습한 머신러닝 모듈을 가지고 있으며 Object Detection 모델로는 Yolo v3를 사용합니다. 주요 함수들은 다음과 같습니다.

1) ml/image : 사진 파일을 서버로부터 받아, 머신러닝 모델에 detection 시키는 함수로, detection이 완료되면 결과를 중앙서버로 전달합니다.

2) ml/learn : 머신러닝에 학습시키기 위한 이미지를 머신러닝 서버에 저장하는 함수로, 이미지는 실제 production에서 사용하는 이미지를 저장합니다. 또한 실제 켄넬안의 상황과 토사물을 학습하기 위해, 미리 유사한 환경을 구성하여 이미지를 생성하고, 해당 함수를 통해 머신러닝 모델에 저장하였습니다.

- **스마트 홈 기기의 Raspberry Pi 4B :**

  1) Bangul_Detect.py : Tensorflow Lite를 사용해서, 비디오 스트림에서 강아지를 검출하고, 강아지의 픽셀 좌표를 반환하는 역할을 합니다. 반환된 값을 Dog_Tracking_Code로 전송해 객체에 대한 판단을 한 뒤, 결정된 회전 값을 Motor_Rotate로 보내 강아지의 위치에 맞는 회전을 시행할 수 있도록 합니다.

  2) Dog_Tracking_Code.py : 반환된 강아지의 픽셀 좌표를 기준으로 현재 카메라로부터 강아지의 좌표를 측정, 이후 추적 및 매칭 알고리즘을 통해 이전에 존재하던 객체와 동일한 객체인지, 새로 만들어진 객체인지 판단합니다. 후에 설정된 강아지의 각도를 반환합니다

  3) Motor_Rotate.py : 모터의 현재 각도와 회전할 각도를 기준으로, GPIO control 과 서보 모터에서 선형보간을 통한 계산을 통해 정해진 회전 각으로 모터를 회전시킵니다.

- **스마트 홈 기기의 Raspberry Pi 3B+** :

  UV4L을 설치 후 실행해, 영상을 차량용 인포테인먼트 기기에 스트리밍 할 수 있도록 합니다.

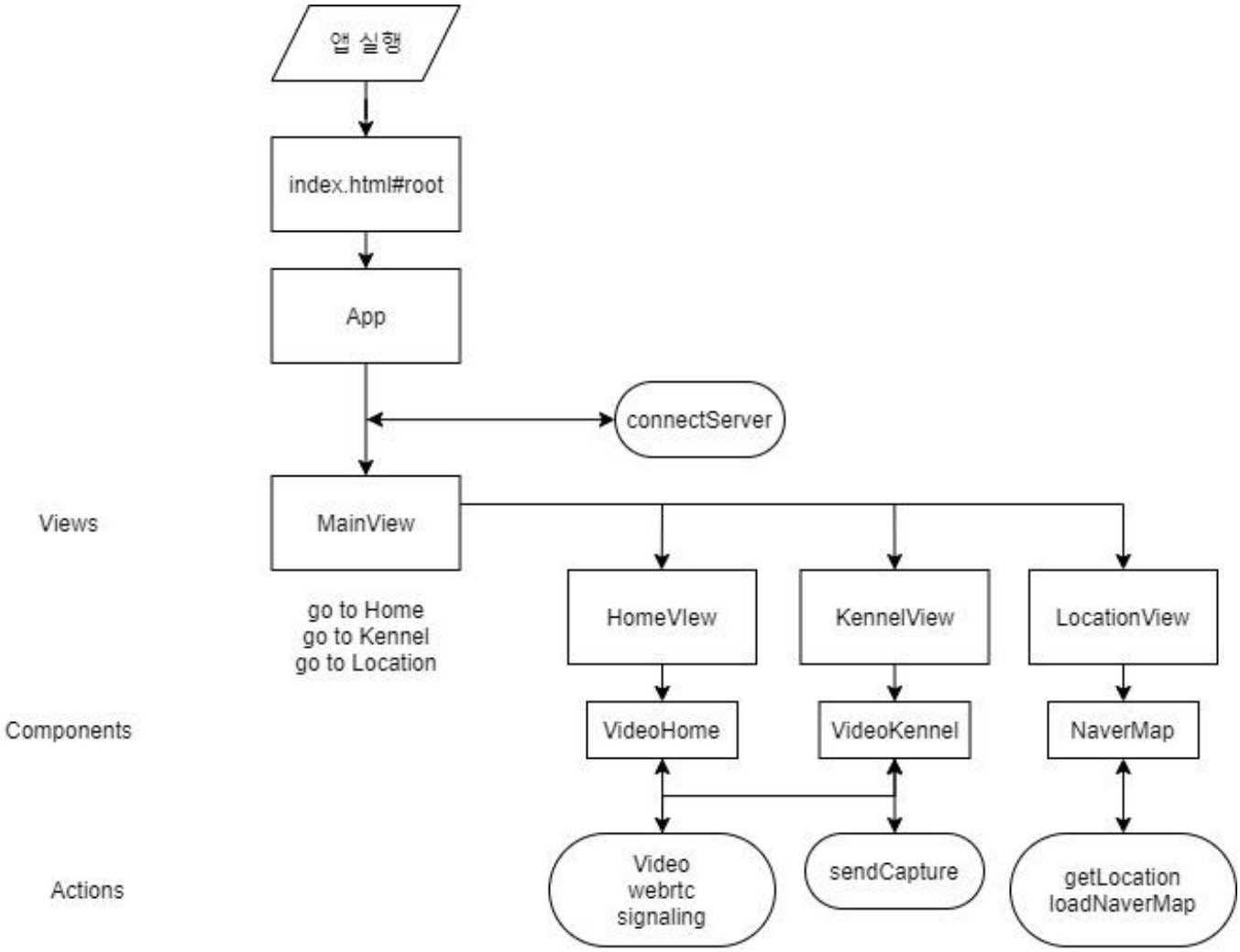
- **스마트 켄넬 기기 :**

  1) UV4L을 설치 후 실행해, 영상을 차량용 인포테인먼트 기기에 스트리밍 할 수 있도록 합니다.

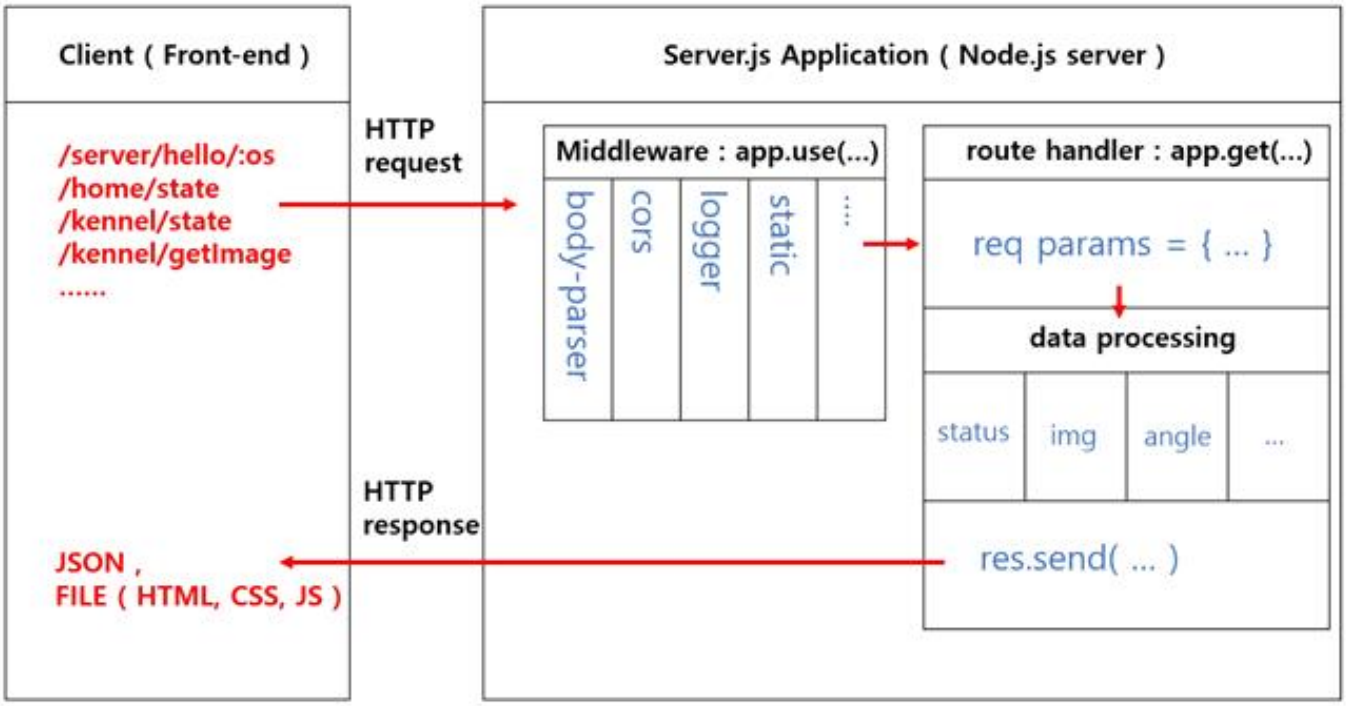
  2) Dht22.py : 켄넬에 설치된 DHT 센서를 GPIO contorl을 통해 읽어와 현재 온도 및 습도를 반환합니다.

○ **Software 설계도 (흐름도 및 클래스 다이어그램 등 / 개발언어에 따라 선택)**

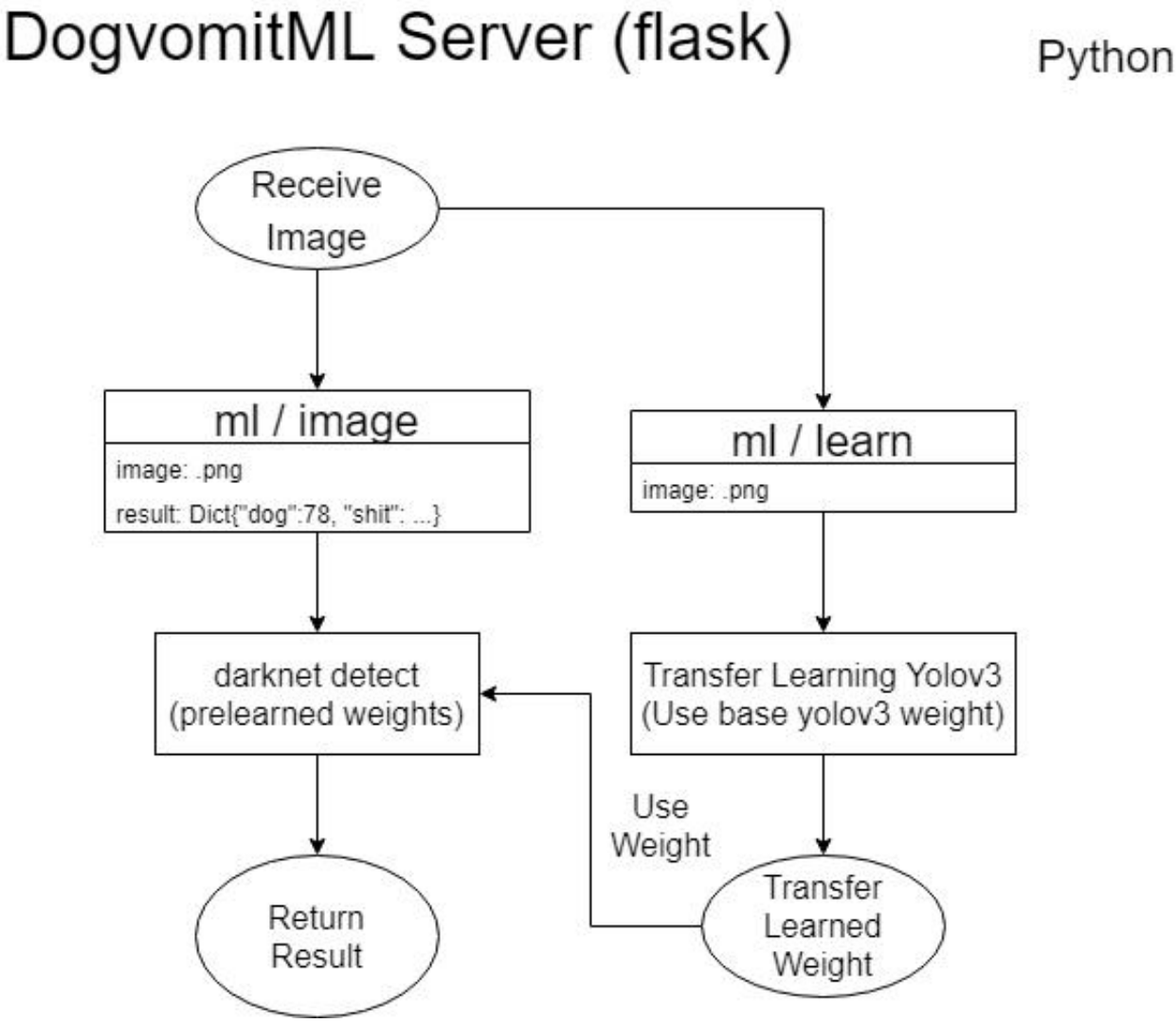
- 차량용 인포테인먼트 기기 :



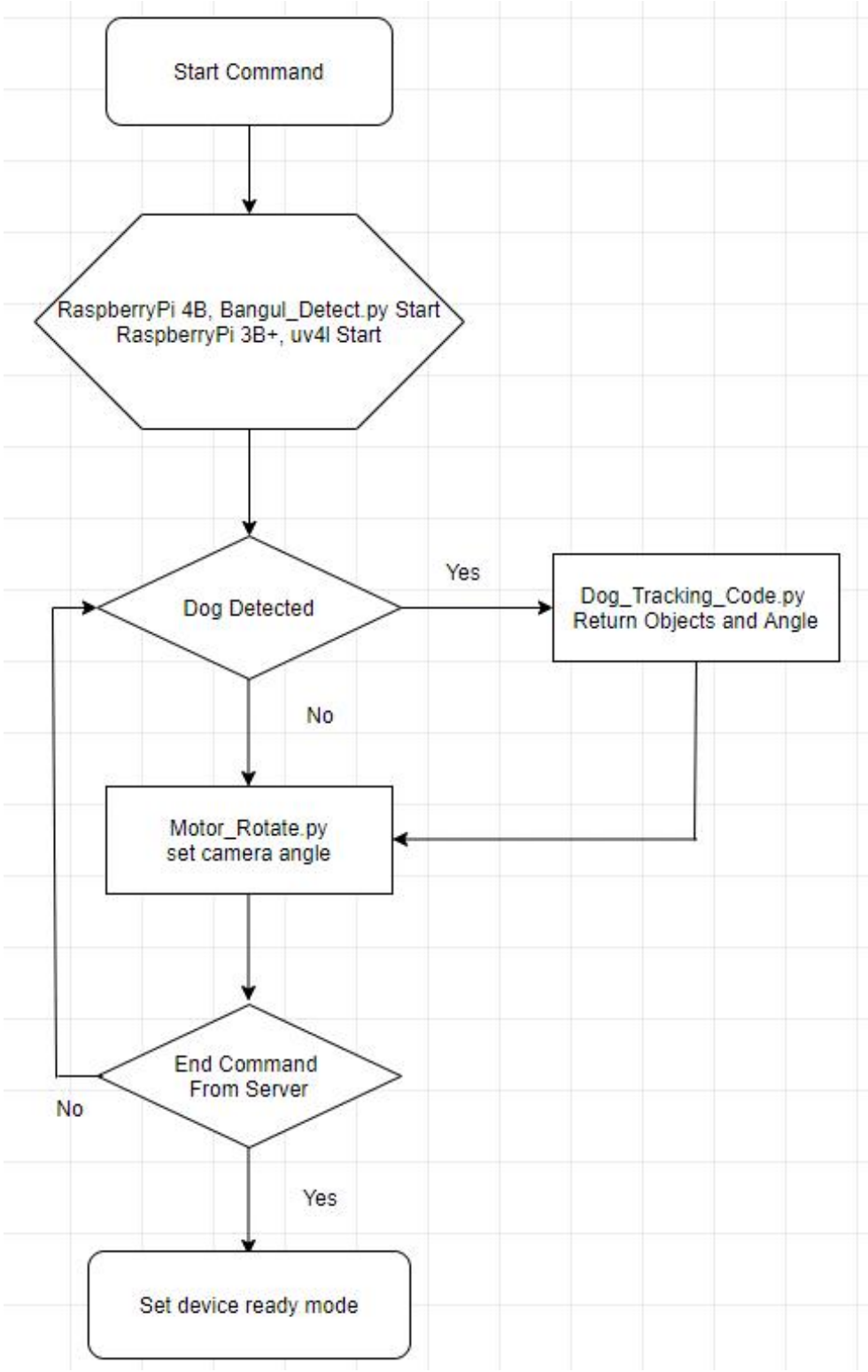
- 메인 서버 :


Client (Front-end)
Server.js Application (Node.js server)
/server/hello/:os
/home/state
/kennel/state
/kennel/getImage
.....
HTTP
request
Middleware : app.use(...)
body-parser
cors
logger
static
....
route handler : app.get(...)
req params = { ... }
data processing
status
img
angle
...
HTTP
response
JSON ,
FILE (HTML, CSS, JS)
res.send(...)

- **DogVomitML 서버 :**



- 스마트 홈 기기 :


Start Command
RaspberryPi 4B, Bangul_Detect.py Start
RaspberryPi 3B+, uv4l Start
Dog Detected
Yes
Dog_Tracking_Code.py
Return Objects and Angle
No
Motor_Rotate.py
set camera angle
End Command
From Server
No
Yes
Set device ready mode

○ **Software 기능 (필요 시 알고리즘 설명 포함)**

- 차량용 인포테인먼트 기기 :

1) Video 기능

- 영상음성 통신 기능을 사용하기 위해서 Video.js, webrtc.js, signaling.js 가 필요합니다.
- Video.js에서 Component가 생성되었을 때, webrtc.js에서 WebrtcSession Class를 이용하여 session을 생성합니다.
- session의 Callback 함수들을 이용하여 연결 및 설정이 완료 되었을 때 동작을 정의합니다.

ex) setOnStreamCallback : stream이 연결되었을 때
setOnDataChannelCallback : dataChannel이 연결되었을 때
getUserAudio : 장치에서 음성장치를 찾았을 때

- 설정을 완료한 후 session.call 함수를 이용하여 통신을 구현합니다
- call 함수에서는 RTCPeerConnection의 인스턴스인 pc를 이용하여 ICE 서버, stream 추가, datachannel 추가를 실행합니다
- 또한 signaling.js에서 받은 signalingChannel을 이용하여 연결 상태 신호를 주고받습니다.

2) Vomit 기능

- KennelVideo에서 remoteVideo에 있는 영상을 가상의 canvas로 캡쳐합니다. 캡쳐된 사진은 서버로 전송하고, 이를 10초마다 반복합니다.

3) 지도 기능

- 앱이 실행되었을 때 head에 네이버 API 사용을 선언합니다.
- LocationView로 들어가면 서버에서 받은 네이버 지도정보를 띄워주고, getLocation Action을 사용하여 현재 장비의 위치를 파악합니다.

4) createToast 기능

- 서버/인터넷/음성장비 연결 불량일 때와 강아지가 토했을 때 webOS API를 이용하여 알람을 띄워줌
- createToast Action이 실행되면 LS2Request를 이용하여 luna2 bus에 createToast method를 보낼 수 있음.

- 메인 서버 :

1) RESTful API 제공
- 서버상태, 홈 디바이스 상태**,** 카메라 이미지 데이터 등 서버 자원을 프론트 엔드에 제공하는 목적으로 사용됩니다.

2) 지도 서비스
- 네이버 지도 API 사용하여 현재 위치 기반 근처의 애견카페, 식당, 공원 등의 장소 정보를 제공합니다.

3) SSL 적용
- 스트리밍 서비스와 같이 민감한 정보에 대한 보안성을 높이기 위해 SSL 인증서를 적용하여 https를 사용하여 데이터 교환합니다.

4) 서버 로그 기록
- 서버 접근 기록을 날짜별로 파일로 보관하기 위해 Winston 모듈 적용하여 구현. 홈 디바이스, 차량용 인포테인먼트 기기와의 데이터 송,수신 내역을 보관합니다.

- DogVomitML 서버 :

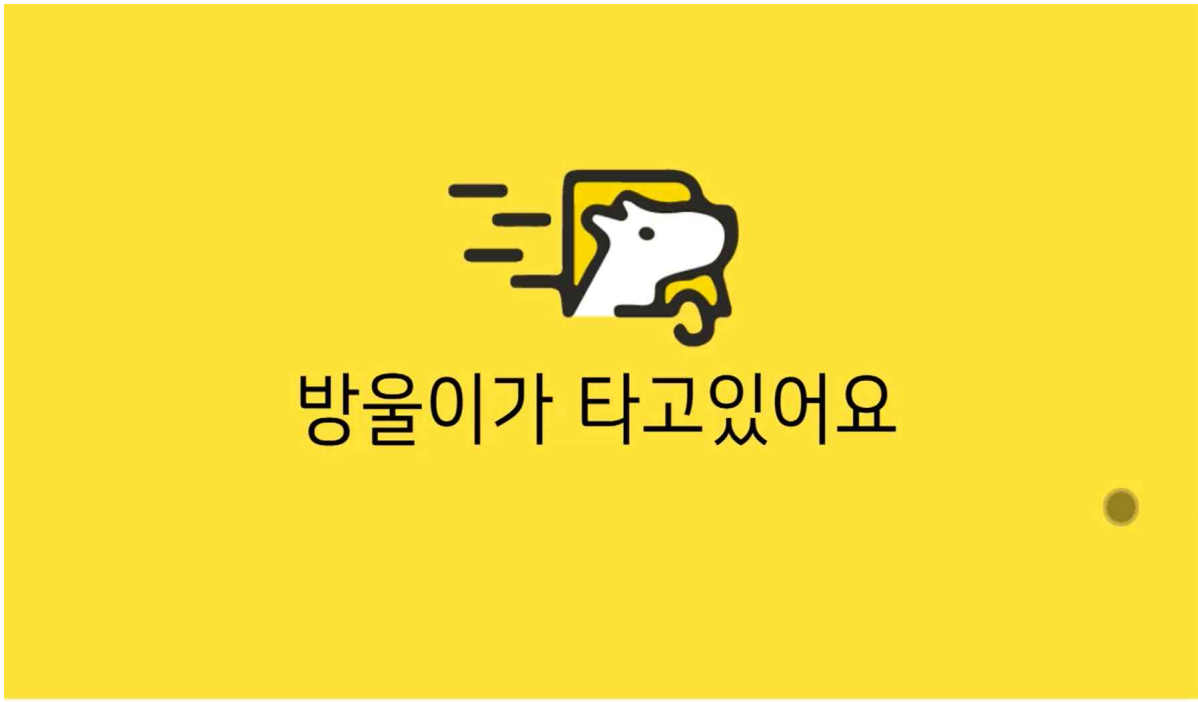
1) 토사물 판단 알고리즘 :
- 멀미하는 사진(입을 벌리고 있는 사진, 귀를 젖히고 있는 사진 등)과 일반적인 사진, 두 분류의 강아지 사진 데이터 세트를 학습한 머신러닝 YOLO v3 모델을 서버에 설치하여 화경을 구성합니다.
- DogVomitML 서버가 메인 서버에서 보내주는 이미지를 받으면, 이미지 검출을 하여 사진을 판단, 결과 값을 다시 메인 서버로 전달합니다. 이때 결과값은 개와 토사물을 중심으로 출력합니다.
- 토사물의 판단 기준은 이미지의 판단 결과가 임계값을 넘었을 경우로 합니다.

- 스마트 홈 기기 :

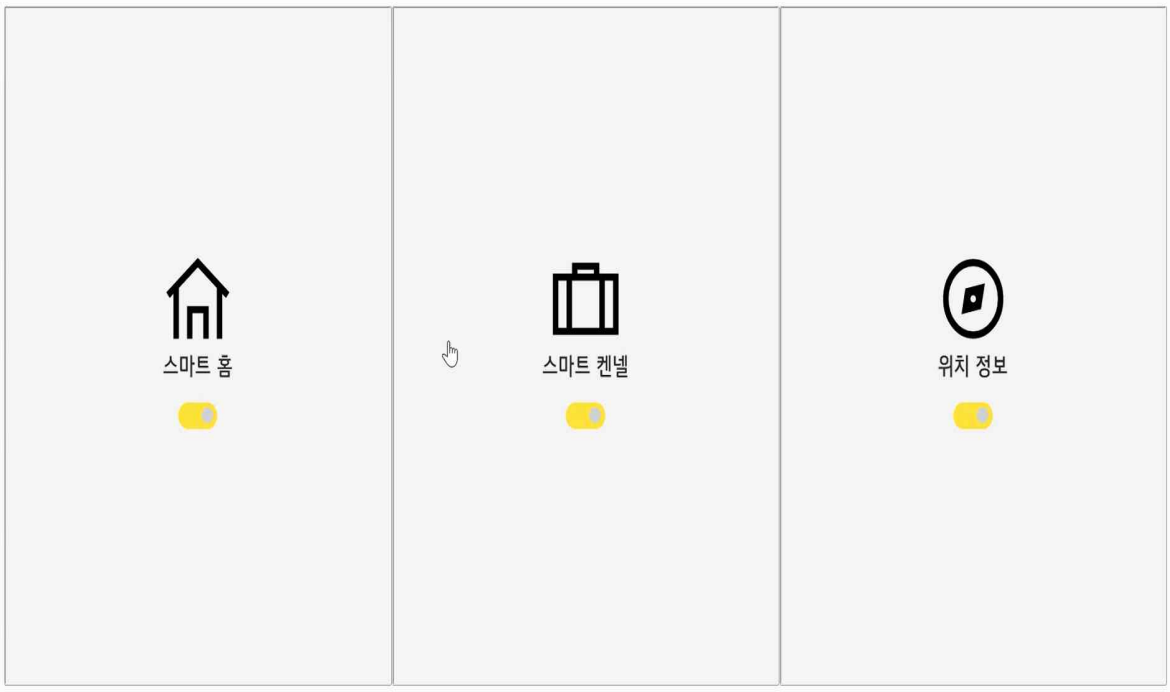
1) 객체의 Depth 구하기 :
- 객체의 Depth의 경우 GML Camera Calibration Tool을 통해 구한 Principal Point와 Focal Length를 활용하여, 삼각법을 통해 구했습니다.

2) 추적 알고리즘 :
- 추적 알고리즘의 경우 예측지점 생성으로는 확장형 칼만필터와 직전 존재 지점의 혼합을 통한 예측지점 생성, 매칭 알고리즘으로는 구해진 예측지점과 측정지점 간의 거리값, 그리고 ROI의 색상에 대한 코사인 유사도를 이용해 헝가리안 알고리즘을 통한 매칭으로 객체의 동일 여부 혹은 생성 및 소멸 여부를 판단하였습니다.

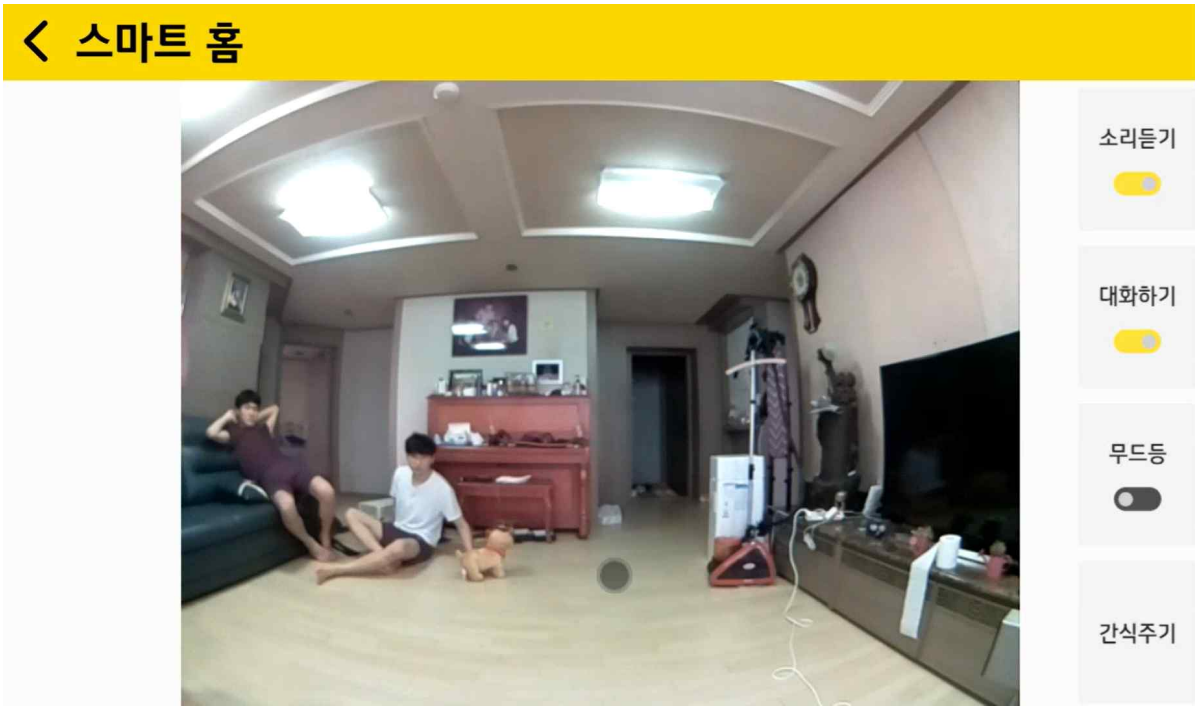
○ UI 사용법 (구성 및 설명)



[로딩화면]

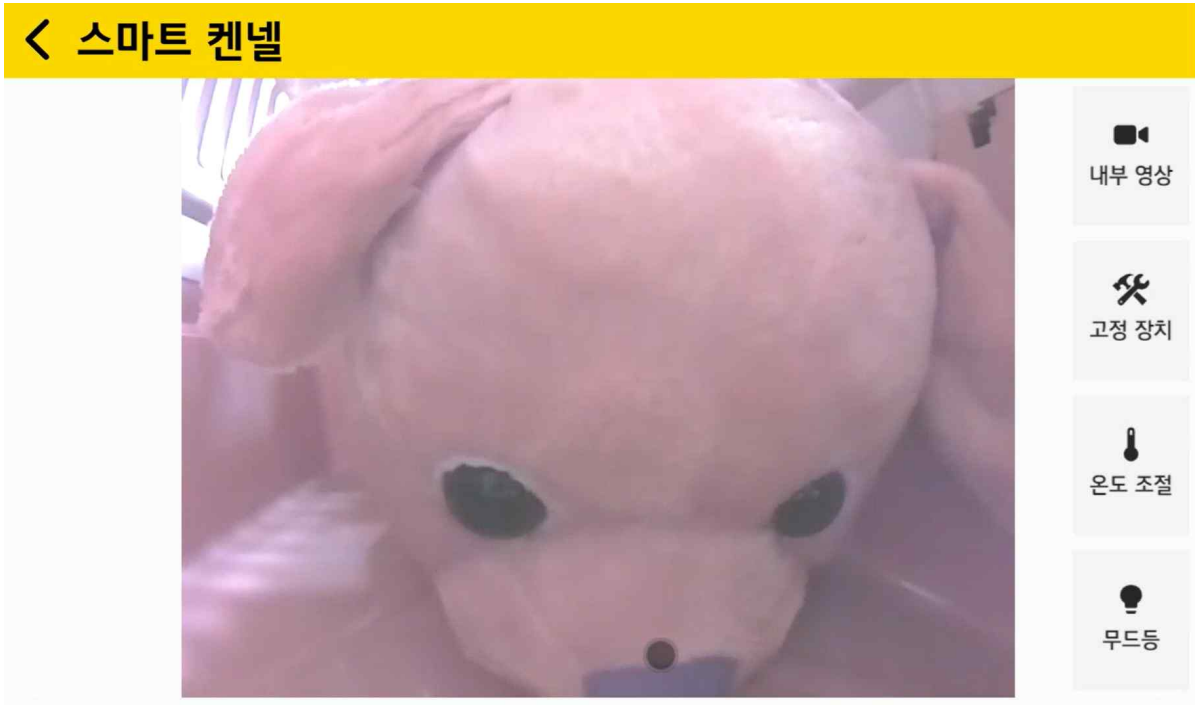


[메인화면]



[스마트 홈 기능]

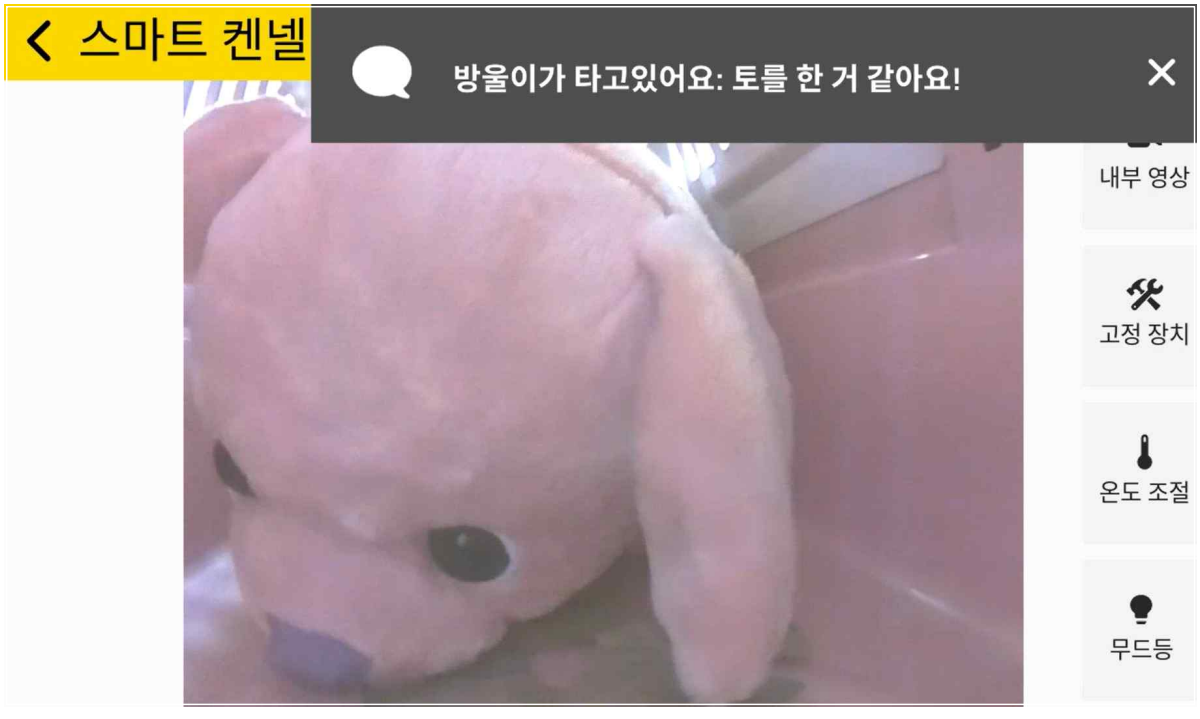
- 집에 있는 반려견을 카메라를 통해 볼 수 있음.
- 소리듣기 on/off 전환 가능
  - on일시, 스피커로 반려견의 소리를 들을 수 있음.
- 대화하기 on/off 전환 가능
  - webos에 내장되있는 마이크를 통해 집에 연결된 스피커로 소리가 나옴
- 무드등 on/off 전환 가능
  - 집에 혼자 있는 반려견의 외로움을 달랠 수 있음
- 간식주기
  - 버튼을 누르면 집안에 간식기기에서 반려견에게 일정량의 간식을 배분해줌



[스마트 켄넬 기능]

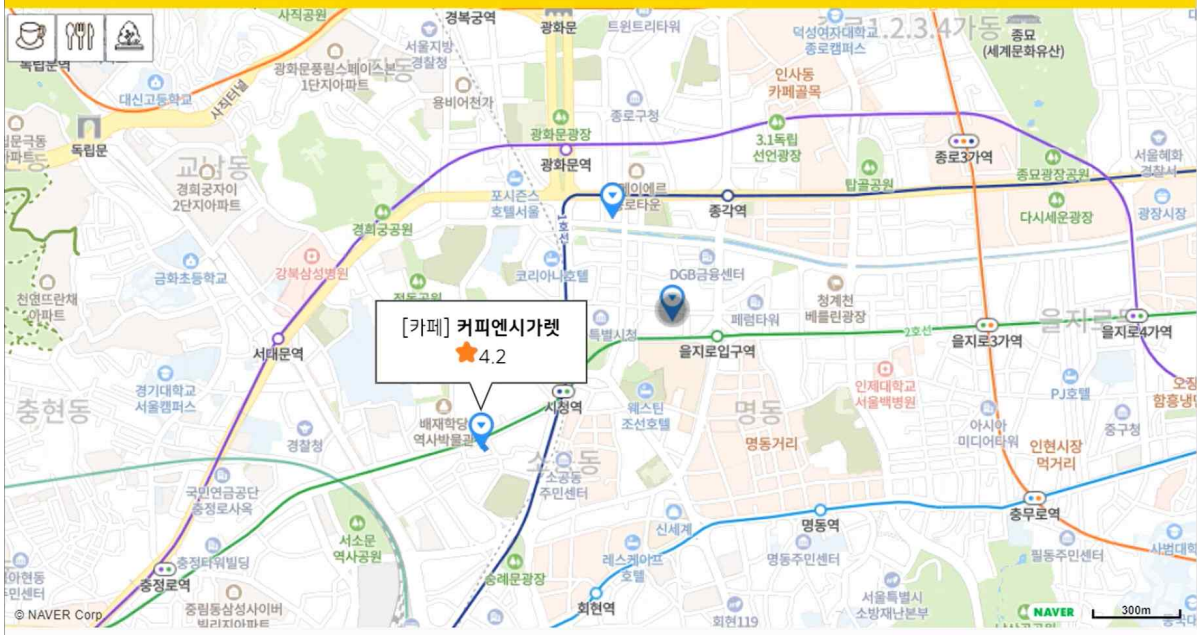
-캔넬 안의 상황을 카메라로 보여준다

- 내부 영상
  - 캔넬 안의 카메라 화면을 전체화면으로 바꿔줌
- 온도 조절
  - 캔넬 안의 온도를 조절할 수 있음
- 무드등
  - 캔넬 안의 무드등의 밝기를 조절할 수 있음



- 캔넬 카메라에 토가 인식되면, 자동으로 푸쉬 알림을 보냄.

위치 정보



[위치정보]

- 맵 왼쪽 위에 아이콘을 누르면 주변의 카페/음식점/공원이 검색됨
- 지도의 아이콘이 표시되면 해당 맵의 아이콘을 클릭하여 위치까지의 경로를 표시함

○ 개발환경 (언어, Tool, 사용시스템 등)

- 차량용 인포테인먼트 기기 : JavaScript, enactjs, moonstone, react, redux, webrtc-adapter, axios

- 메인 서버 : Node.js AWS EC2, Node.js Modules : request, body-parser, Winston, express, https

- DogVomitML 서버 : Python 3.6.11, AWS EC2, OpenCV, Python Machine-Learning Library(numpy, pandas, tensorflow, scipy, tensorborard, scikit-learn)

- 스마트 홈 기기 : Python 3.6.11, OpenCV, TFLite, Python GPIO Library

- 스마트 켄넬 기기 : Python 3.6.11, Python GPIO Library

□ 개발 프로그램 설명

○ 파일 구성

- 차량용 인포테인먼트 기기 :
  - webos-meta : webos용 App을 만들 때 필요한 정보를 담습니다.
    - resources : 사진, 텍스트 등 App에 필요한 자원을 저장한 곳입니다.
    - src : App을 구성하는 소스코드
      - actions : 상태를 저장하기 위한 동작들을 모아둔 곳입니다.
        checkAction, connectAction, locationAction, videoAction, webosAction
      - api : 서버와 통신하기 위한 모듈입니다.
      - App : App이 실행되고 #root를 통해 가장 먼저 실행되는 Component입니다.
      - components : 디자인과 기능을 구성하는 기본 단위입니다.
        - Box : 창과 버튼 등을 배열하기 위해 사용됩니다.
        - Button : 많이 사용되는 버튼의 디자인과 기능을 정의합니다.
        - Common : 공통적으로 사용되는 Icon, Switch 등이 있습니다.
        - Home : Home에서만 사용되는 BtnTalk, VideoHome이 있습니다.
        - Kennel : Kennel에서만 사용되는 VomitCheck 등이 있습니다.
        - Location : 위치 정보를 나타내기 위해 사용됩니다.
        - Main : 앱을 실행시키자마자 나오는 로딩화면, 메인화면을 구성합니다.
        - Video : 영상통신을 구현하기 위한 구성요소들을 포함합니다.
      - reducers : action으로부터 신호를 받아 store의 상태를 변화시킵니다.
      - store : redux의 상태를 저장하는 공간입니다.
      - views : component로 구성된 화면을 보여줍니다.
        HomeView, KennelView, LocationView, MainView

- 메인 서버 :
  - Bangul_Smart -> server.js : 메인 서버 실행 파일
    -> main -> middleware -> winlogger.js : 서버 로그 라이브러리 파일
    -> naverMap.html : 네이버 지도 html 파일
    -> naverMap.js : 네이버 지도 API 파일
    -> pfx -> certificate.pfx : SSL 인증서 파일

- DogVomitML 서버 :
  - dogvomitML Server ->
    - dogvomitML/server.py : 서버를 실행시키는 메인 파일
    - TrainYourOwnYOLO/Training (Directory): 서버를 실행시키는 메인 파일
    - TrainYourOwnYOLO/Data/Model_Weights (Directory): yolov3를 학습시키는 소스파일

- 스마트 홈 기기 :
  - src -> tflite :
    - tflite1-env : 라즈베리파이 상에서 tflite를 돌릴 수 있는 가상환경으로 돌입할 수 있게 해줍니다.
    - Sample_TFLite_model : 강아지를 포함한 다양한 모델이 들어있는 폴더입니다.
    - Bangul_Detect.py : tflite를 활용해 객체를 감지하고 판단을 내리는 파일
    - Dog_Tracking_Code.py : 각각의 객체에 대한 정보를 갱신 및 생성, 삭제하는 파일
    - Motor_Rotate.py : 촬영용 카메라가 부착된 모터를 회전시키는 파일

- 스마트 켄넬 기기 :
  - src :
    - DHT22.py : 온습도 센서에서 값을 읽어와 서버로 전송하는 파일

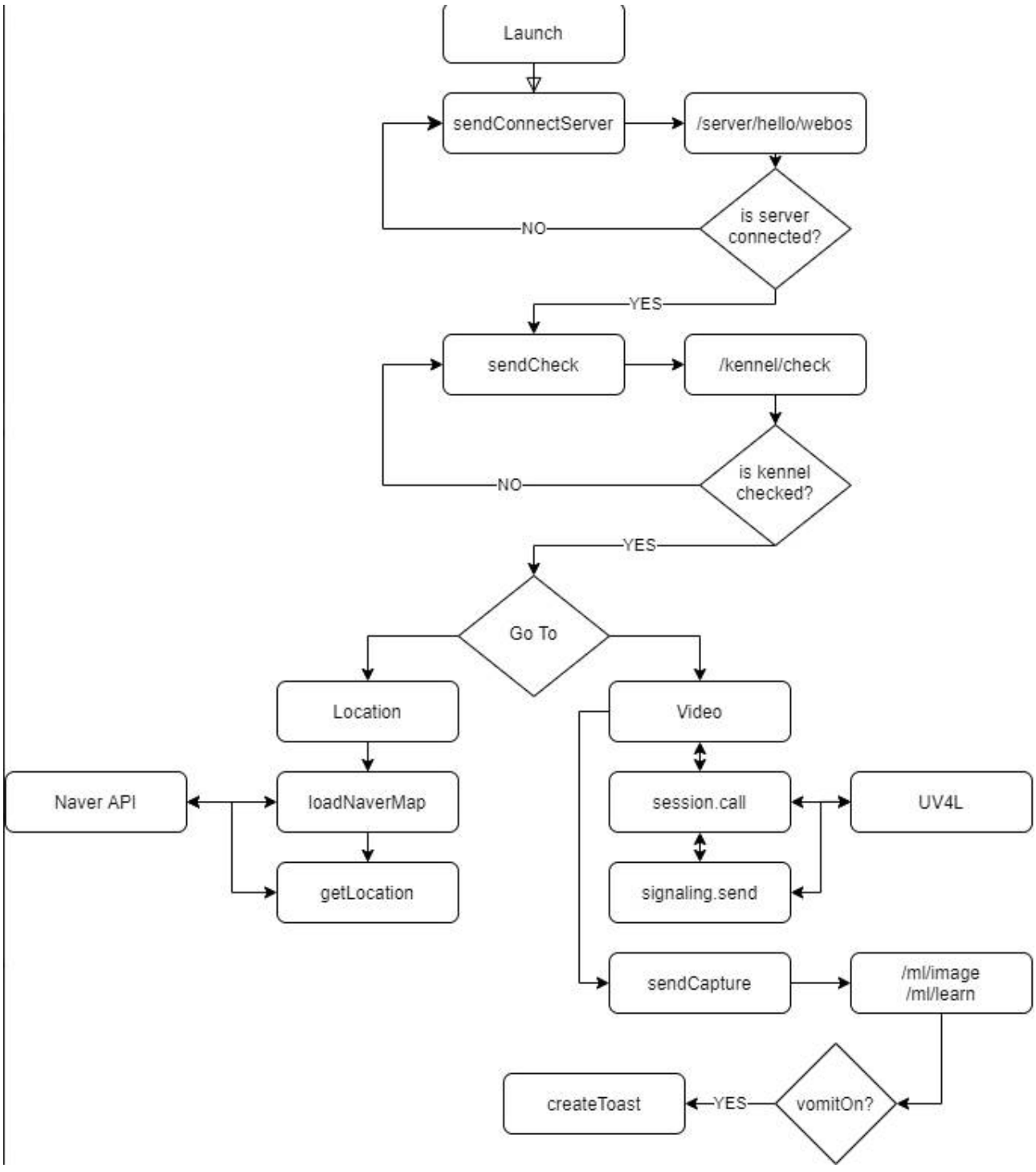
○ 함수별 기능

- 차**량용 인포테인먼트 기기와 메인 서버, DogVomit 서버로 이어지는 함수** :
    - sendConnectServer : 서버에 GET /server/hello/webos을 요청합니다. 응답상태가 200이면 serverOn을 true로 하고, 오류가 발생하면 createToast로 알람을 보냅니다.
    - sendCheck : 서버에 GET /kennel/check, /home/check를 보냅니다. 응답상태가 200이면 isOn을 true로 바꿉니다. location에 경우 loadNaverMap() 함수를 실행합니다.
    - sendCapture : 영상을 캡쳐한 이미지를 base64 형태로 서버에 보냅니다. 이때 POST /ml/image, /ml/learn을 이용하여 보냅니다. {dog, shit}의 반환값을 이용하여 반려견의 상태를 파악합니다.
    - loadNaverMap : 네이버 API를 사용하여 지도를 받습니다.
    - getLocation : 네이버 API를 사용하여 네트워크 기반으로 현재 위치를 받습니다.
    - createToast : webOS API를 사용하여 webOS용 알람을 보냅니다. enactjs의 라이브러리 LS2Request를 사용했습니다.

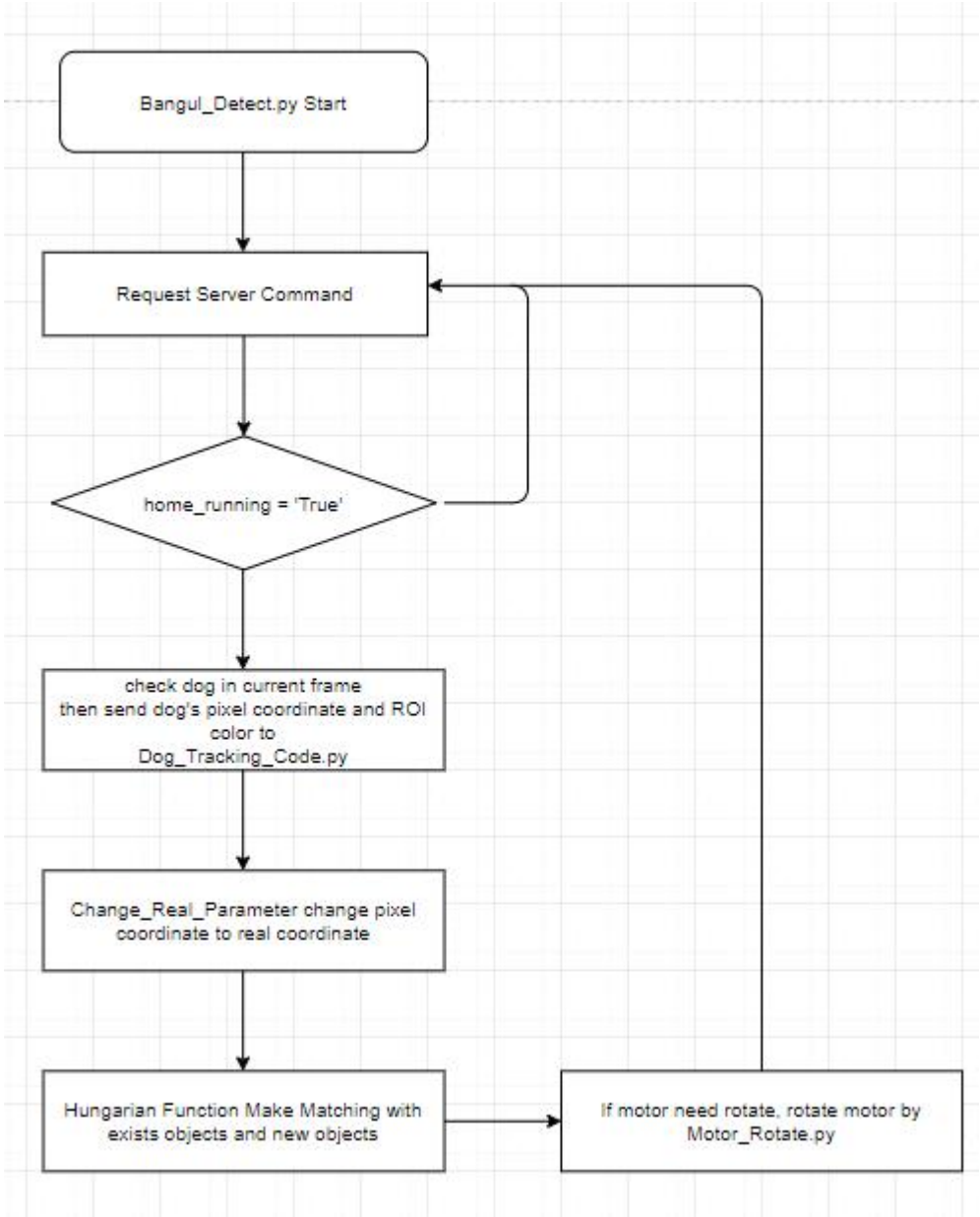
- **스마트 홈 기기의 Dog Tracking Code.py 내 함수** :
    - Change_Real_Parameter : 픽셀 좌표계의 값을 칼리브레이션 값을 토대로 현재 기기로부터 얼마나 떨어진 위치인지 확인해줍니다.
    - Hungarian : 측정지점과 매칭지점을 O(n^4)의 연산을 통해 Optimal Matching으로 묶어줍니다. Konig 함수를 통해 매칭 체크를 시행합니다.

○ 주요 함수의 흐름도

- **차량용 인포테인먼트 기기와 메인 서버, DogVomit 서버의 흐름** :



- 홈 IoT기기와 메인 서버간의 흐름 :



○ 기술적 차별성

- yolov3 기반 토사물 detection 머신러닝 모델
    - 기존 실시간 이미지 인식이 좋은 yolov3의 모델에 캔넬안의 강아지와 토사물을 학습시켜, 실제 자동차 안의 캔넬 환경의 object detect에 강하게 설계하였습니다.
    - 기존 yolov3를 기반으로 하여 속도가 빠르고 detection하는데 자원이 많이 들지 않아, 효율적입니다.
    - 캔넬안에서 다른 머신러닝 모델들에 비해, 강아지와 토사물 인식률이 좋습니다.

- SSL 적용을 통한 보안성 향상
    - 개인 프라이버시와 연관된 카메라 이미지 데이터를 https 프로토콜을 사용하여 통신함으로써 보안성을 높였습니다.
    - SSL을 적용함으로써 지도 서비스에 필요한 HTML5 geolocation (사용자의 현재 위치 제공) 을 사용하여 현재 위치 기반의 장소 추천 서비스를 구현하였습니다.

- WebRTC를 이용한 영상, 음성 스트리밍
    - 지연 없는 스트리밍으로 강아지의 상태를 실시간으로 확인할 수 있습니다.
    - 플러그인의 도움 없이 스트리밍할 수 있습니다.

- 추적 알고리즘 설계
    - 강아지의 자연스러운 추적을 위해 TPU장치를 사용해 연산의 속도를 높였습니다.
    - 추적의 정확도를 높이기 위해 예측지점은 칼만필터와 현재 지점을 혼용하여 생상하고, 매칭알고리즘에서 ROI Color와 예측점과 측정지점 간의 거리를 코사인 유사도를 통해 적정한 비율로 혼합해, 높은 수준의 매칭 알고리즘을 보유해, 다수의 강아지가 있더라도 원하는 강아지를 추적할 수 있습니다.

□ 개발 중 장애요인과 해결방안

○ webOS 개발 시 장애요인

- moonstone 스킨 색상 변경
  - moonstone 라이브러리를 바로 가져와서 직접 style안에 color.less를 수정하는 방법으로 문제를 해결하였습니다.
- 서버와 통신 시, 주기적인 신호 주고받지 못함.
  - 주기적으로 신호를 주고받기 위해 setInterval과 async, await을 사용하여 문제를 해결하였습니다.

○ 서버 구축 시 장애요인

- 데이터 프라이버시 보호의 필요성
  - 카메라 데이터의 보안성 향상을 위해 https를 사용하였습니다.
  - https를 사용하여 송수신하기 위해서 SSL을 사용하여 문제를 해결하였습니다.
- 협업 개발 시 서버 상태 확인의 어려움
  - 서버가 수신하는 HTTP 요청 기록을 보관하기 위해 winston 모듈을 적용해서 로그를 일자별로 나누어 저장하였습니다. 이를 통해, 개발자들이 개발 중 서버 상태를 자유롭게 확인할 수 있게 되었습니다.

○ 머신러닝 부분 장애요인

- 초기에 토사물 사진을 구하기 어려움
  - 이를 해결하기 위해, 몇 개의 토사물 사진을 데이터 변조(agumentation) 시켜 여러장의 토사물 데이터 세트를 만들었습니다.
  - Crop, Flip, Translate, Resize등 다양한 방법으로 변조시켰습니다.
- 여전히 토사물 사진 데이터가 부족
  - 이를 해결하기 위해 실제 캔넬에서 토사물과 강아지를 가지고 이미지를 자체생성하여 데이터 세트를 확보하였습니다.
  - 실제 사용될 시, 사용된 사례에서 데이터 세트를 확보하였습니다. (보안 문제는 위 기술된 방법으로 해결)

○ 강아지 추적 시 장애요인

- 낮은 연산속도로 자연스러운 추적 불가
  - 구글 코랄 USB을 이용하여 행렬 연산속도를 높여 자연스러운 추적이 가능하도록 해결하였습니다.

○ 영상 스트리밍 부분 장애요인

- 음성 송수신을 위해 webOS에 pyaudio와 같은 모듈 설치 과정에서의 문제 발생
  - WebRTC를 이용하여 영상, 음성 송수신을 해결하였습니다.
- WebRTC를 사용하기 위한 UV4L 설치 과정 중 에러 발생
  - UV4L은 라즈비안 Stretch 버전까지 설치할 수 있어 Buster버전에서 Stretch버전으로 다운그레이드하여 설치를 완료하였습니다.

□ 개발결과물의 차별성

○ 차량용 기기를 통한 반려견 케어

- 차량용 기기를 통해 홈 기기와 켄넬 네트워크에 대한 접근이 가능하여 차량에서도 쉽게 반려견을 관리할 수 있습니다. 반려견을 자동 추적해주는 카메라로 반려견의 현재 모습을 관찰하거나, 음성을 통해 대화할 수 있습니다. 또한 반려견이 켄넬 탑승 시 구토, 멀미가 발생할 것에 대비하여 머신러닝 카메라를 통해 이미지 검출을 통해 상황을 파악하여 반려견의 상태가 양호한지 파악할 수 있도록 합니다.

○ 스마트 켄넬

- 기존에 출시된 켄넬의 역할은 단순히 반려견을 안전하게 보호하며 이동하는 것에 있었습니다. 그러나 이번 개발 작품은 켄넬의 기능을 확장시켜 조명, 카메라, 마이크 등의 모듈을 적용하였습니다. 이런 모듈들을 네트워크를 통해 원격으로 제어할 수 있게하면서 IOT 시대에 더욱 발전된 기술을 적용한 켄넬을 제시합니다.

○ 반려견, 반려인의 활동 반경 확대

- 기존에는 반려견을 데리고 외출 하는 것에 대한 어려움이 많았습니다. 그러나 켄넬과 지도 서비스 이용을 통하여 반려견과 동반 입장이 가능한 장소로 손쉽게 이동할 수 있습니다.

○ 지도 API를 이용하여 반려견과의 활동 범위 증가

- 반려견과 함께 이용할 수 있는 장소를 지도에 표기함과 동시에 길 찾기 서비스를 제공함으로써 활동 범위를 증가시킬 수 있습니다.

□ 개발 일정

No	내용	2020年			
		6月	7月	8月	9月
1	계획서 작성 및 아이디어 조사				
2	차량용 인포테인먼트 기기 개발				
3	메인 서버 개발				
4	머신러닝 서버 개발				
5	스마트 홈 기기 개발				
6	스마트 켄넬 기기 개발				
7	영상 제작				
8	보고서 작성				

□ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	박채운	- 스마트 홈 제작 - 강아지 추적 시스템 알고리즘 구현 - 라즈베리파이 모터 & 센서 제어
2	팀원	이직주	- 프론트엔드 구축 - 스마트 홈/켄넬 기기와 연동 - UV4L 스트리밍 서버로부터 영상/음성 데이터 수신
3	팀원	김도언	- 머신러닝 서버 환경 구축 - 머신러닝 이미지 데이터 세트 확보 - 머신러닝 모델 생성
4	팀원	오창균	- 서버 개발 환경 구축 - 네이버 지도 API를 이용한 위치 기반 서비스 제공
5	팀원	신선종	- 스마트 켄넬 제작 - 라즈베리파이 영상/음성 UV4L 스트리밍 서버 환경 구축