**EDUM 상세설계서**

**(Emergency Detection CCTV**

**Using Machine Learning)**

**문서 정보**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **구분** | **소속** | **성명** | **날짜** | **서명** |
| **작성자** | 한국외국어대학교 | 전진우 | 2018. 10. 31 |  |
| 한국외국어대학교 | 이대홍 | 2018. 10. 31 |  |
| 한국외국어대학교 | 임광효 | 2018. 10. 31 |  |
| 한국외국어대학교 | 권소연 | 2018. 10. 31 |  |
| 한국외국어대학교 | 김준영 | 2018. 10. 31 |  |
| **검토자** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **사용자** |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **승인자** | 한국외국어대학교 | 홍진표 |  |  |

머리말

본 문서는 카메라의 영상에 대한 객체 인식을 통해 대단위 아파트, 상가단지 등과 같은 거주지역에서의 이상 상황 감지 및 알림을 전송하는 EDUM 시스템에 대한 상세 설계를 기술한 것이다.

**개정 이력**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **버전** | **작성자** | **개정일자** | **개정 내역** | **승인자** |
| 1.0 | 이대홍  전진우  김준영  임광효  권소연 | 2018. 11. 13 | 초안 작성 |  |
| **검토자** | 전진우 | | |
| 1.1 | 이대홍  전진우  김준영  임광효  권소연 | 2018. 11. 14 | 초안 수정 |  |
| **검토자** | 전진우 | | |
|  |  |  |  |  |
|  |  | | |
|  |  |  |  |  |
|  |  | | |

**목차**

1. **개요…………………………………………………………………………………………………………………...7**
   1. **목적...……………………………………………………………………………………………………………7**
   2. **범위...……………………………………………………………………………………………………………7**
   3. **관련문서……………………………………………………………………………………………………..…8**
   4. **용어 및 약어………………………………………………………………………………………………….8**
2. **시스템구성………………………………...………………………………………………………………………..9**
   1. **시스템 구성도……………………………………………………………...………………………………...9**
   2. **소프트웨어…………………………………………………………………………………..………………..9**
   3. **하드웨어…………………………………………………..…………………………………………………10**
3. **기능설명……………………………………………………..……………………………………………………10**
   1. **객체인식…………………………………………………………………..…………………………………10**
      1. **프레임 캡처…………………………………………………………………………..………….11**
      2. **머신러닝………………………………………………………………………………..…………11**
      3. **인식종류…………………………………………………………………………..………………12**

**- 쓰러진 사람…………………………………………………………………………………………...12**

**- 쓰레기…………………………………………………………………………………………..……….14**

**- 접근제한구역 침입자……………………………………………………………………………….15**

* + 1. **인식결과………………………………………………………………………………..…………17**
  1. **객체정보처리………………………………………………………………………………..……………...17**
     1. **정보수집…………………………………………………………………………………………..18**
     2. **알림메세지 전송………………………………………………………………………………..19**
     3. **영상제공…………………………………………………………………………………………..21**
     4. **영상저장…………………………………………………………………………………………..23**
  2. **DataBase…………………………………………………………………………………………………….23**
  3. **사용자…………………………………………………………………………………………………………24**
     1. **관리자………………………………………………………………………………………….…..24**
     2. **일반 주민………………………………………………………………………………………….25**

1. **기능동작……...……………………………………………………………………………………………………26**
   1. **Dataflow……………………………………………………………………………………………………..26**
   2. **Sequence Diagram……………………………………………………………………………………….26**

**5. 자체 시험 방법 및 절차……………………………………………………………………………………….27**

**6. 향후 실제 적용 방안……………………………………………………………………………………………28**

**7. 기대효과…………………………………………………………………………………………………………….29**

**8. 세부 추진 계획 및 일정……………………………………………………………………………………….30**

**그림 목차**

**[Figure1] EDUM 시스템 구성도…………………………………………………………………………………………9**

**[Figure2] 시스템 구성 요소1: 객체인식…………………………………………………………………………….10**

**[Figure3] 이미지에 대한 labeling………………………………………………………………………………….….12**

**[Figure4] 머신러닝으로 생성한 custom model의 object detection………………………………………12**

**[Figure5] 머신러닝으로 수행하여 감지한 가상펜스 침입……………………………………………………..15**

**[Figure6] 시스템 구성 요소2: 객체정보처리………………………………………………………………………17**

**[Figure7] 관리자 경고화면 팝업창…………………………………………………………………………………..20**

**[Figure8] SMS전송…………………………………………………………………………………………………………21**

**[Figure9] 시스템 구성 요소3: 영상제공…………………………………………………………………………….22**

**[Figure10] 전체적인 영상제공…………………………………………………………………….…………………..22**

**[figure11] 시스템 구성 요소4: DataBase…………………………………………………………………………..23**

**[Figure12] 시스템 구성 요소5: 사용자……………………………………………………………………………..24**

**[Figure13] 관리자의 Data요청 및 처리 과정…………………………………………………………………….25**

**[Figure14] Dataflow………………………………………………………………………………………………………26**

**[Figure15] 관리자 입장의 Sequence Diagram…………………………………………………………………..26**

**[Figure16] 거주민 입장의 Sequence Diagram…………………………………………………………………..27**

**[Figure17] 수동 영상관제 시간에 따른 위험 탐지율…………………………………………………………..29**

**[Figure18] 추진계획 및 일정…………………………………………………………………………………………..30**

**표 목차**

**[Table 1] 관련문서………………….……………………………………………………………………………………….8**

**[Table 2] 용어 및 약어………….…………………………………………………………………………………………8**

**1. 개요**

본 장에서는 Inception v2 모델을 이용한 객체 인식을 통하여 아파트 단지 내의 이상 상황들을 감지, 신속한 대처가 가능하도록 알림을 발신하는 시스템인 EDUM의 시스템 및 구성, 기능에 대한 총괄개요를 제공한다. 여기서는 ‘EDUM’의 목적과 이용 범위, 정의사항, 참고자료 그리고 본 상세설계서의 개요를 소개한다.

* 1. **목적**

본 장에서는 EDUM 시스템의 기능과 동작을 구체적으로 명시하고 설계하는데 목적이 있다.

* 영상 스트리밍을 통해 로그인한 주민에게 CCTV의 영상을 제공할 수 있다.
* Inception v2 모델을 이용하여 CCTV 영상을 실시간 객체인식 할 수 있으며, 이를 토대로 세 가지의 이상 상황 감지가 가능하다.
* 이상상황 감지 시에 알림이 송신됨과 동시에 CCTV 영상이 송출되고 있던 화면의 팝업창에 해당 CCTV 영상과 객체 인식된 상황의 사진을 송출한다.
* 넘어진 사람이나 쓰레기 무단 투기를 발견 혹은 접근 제한 구역에 침입한 사람을 감지한 경우 SMS와 Web Page를 통해서도 알림 서비스 제공한다.
  1. **범위**

특정 구역 내에서 발생하는 여러가지 상황을 CCTV를 통해 보다 편리한 관리 감독을 제공하는 시스템으로 적용 가능한 범위로는 대단위 아파트 단지, 학교, 상가 등이 있다

* 1. **관련문서**

|  |  |
| --- | --- |
| **문서** | **문서 제목** |
| 연구성과 실용화 진흥원 | 영상 감시 시스템 시장 및 기술동향 |
| 한국지역정보개발원 | 지능형 CCTV 기술 현황 및 활용 사례 |
| 한국디지털CCTV연구조합 | 차세대 지능형 CCTV 산업 경쟁력 강화 방안 연구 |
| 전자부품 연구원 | 지능형 CCTV 시스템 기술 이슈 및 산업동향 |
| Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun | Faster R-CNN: Towards Real-TimeObject Detection with Region Proposal Networks |
| Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik UC Berkeley | Rich feature hierarchies for accurate object detection and semantic segmentation |

**[Table 1] 관련문서**

* 1. **용어 및 약어**

|  |  |
| --- | --- |
| 용어 및 약어 | 풀이 |
| COCO dataset | Common object in Context dataset |
| SMS | Short Message Service |
| HTML | Hyper Text Markup Language |

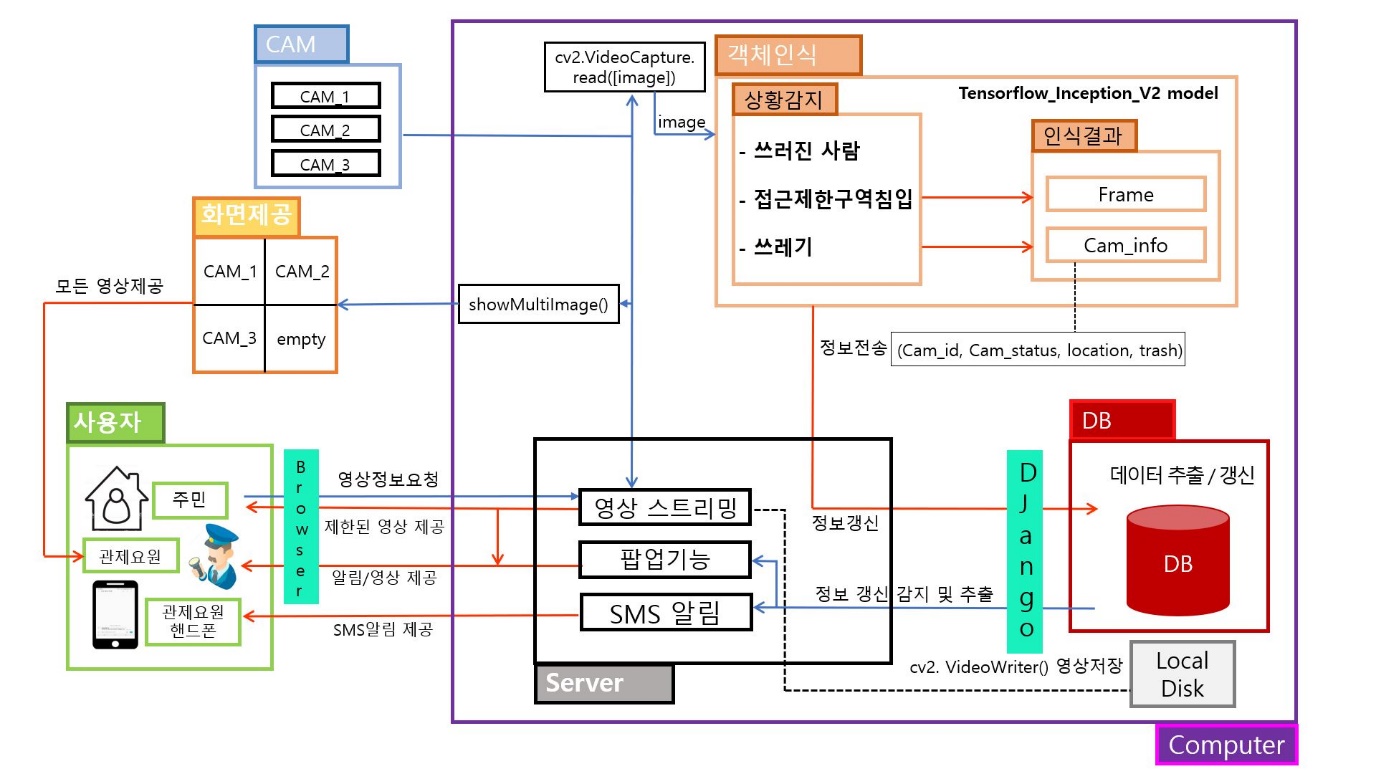
**[Table 2] 용어 및 약어**

* Faster RCNN: Fast R-CNN의 Region Proposal 방법인 Selective Search 방식을 개선한 방식으로CNN을 통해 추출된 특징 맵을 RPN에 입력한다. RPN(Region Proposal Networks)에 입력 시 Object가 있을 만한 구역에 대한 Proposal을 연산한다.
* 가상 펜스: 접근 제한 구역에 가상으로 그어진 선으로 이 선을 넘어가는 사람을 감지하면 관리자에게 알림을 보낸다.

**2. 시스템구성**

**2.1 시스템 구성도**

**‘EDUM’ 시스템의 구성은 아래 그림과 같다.**

****

**[Figure1] EDUM 시스템 구성도**

**2.2 소프트웨어**

* **Object detection**

OpenCV를 통해 프레임을 받아 Inception V2 model을 사용하여 사람, 쓰레기, 누운 사람을 인식한다.

* **DataBase**

Django에서 제공하는 Sqlite3를 이용하여 서버와 연결하여 사용한다.

* **Server**

Django를 이용하여 웹서버를 제작하고 Daphne를 사용하여 웹서버를 배포한다.

* **UI**

웹 인터페이스는 HTML을 이용하여 페이지를 작성한다.

* **SMS**

CoolSMS에서 제공하는 API를 통하여 SMS을 사용자에게 전송한다.

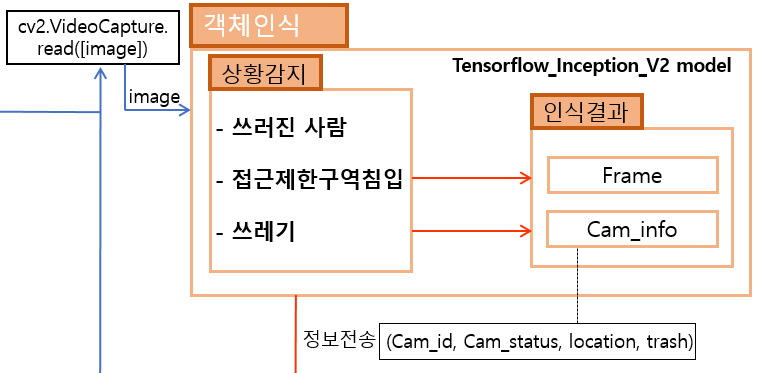
**2.3 하드웨어**

* **웹 카메라**

두 대의 웹 카메라를 객체인식 및 서버용 컴퓨터에 연결, 이를 CCTV로 구현하기 위해 압축 저장과 분할된 화면 송출을 구현한다.

**3. 기능설명**

**3.1객체인식**

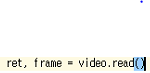


**[figure2] 시스템 구성 요소1: 객체인식**

본 장에서는 카메라에서 제공하는 영상을 5초에 한 번씩, OpenCV를 사용하여 한 프레임을 캡처하여 기존 Inception V2 model에 추가 학습한 custom model을 사용하여 객체를 인식하는 객체 인식 시스템을 서술한다. 시스템은 프레임 캡처, 머신 러닝, 인식 종류, 인식 결과로 나누어 상세히 서술한다.

**3.1.1 프레임 캡쳐**

5초마다 OpenCV의 VideoCapture 클래스를 사용하여 영상을 캡처하는데cv2.VideoCapture() 메소드를 사용한다. 성공하면 <VideoCapture object>를 반환한다. 이를 video 변수에 저장하며 해상도는 640\*360로 설정한 후 video.read([image]) 메소드를 사용하여 image를 객체 인식에서 가공할 이미지로 사용한다.

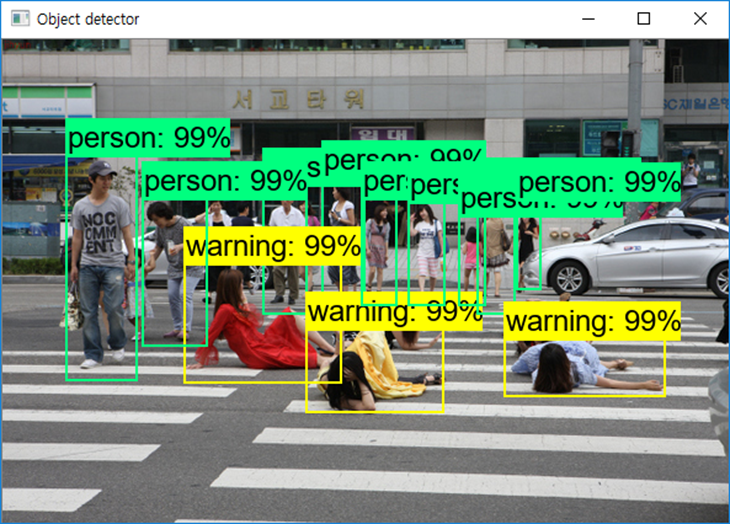


**[프레임 초기설정]**

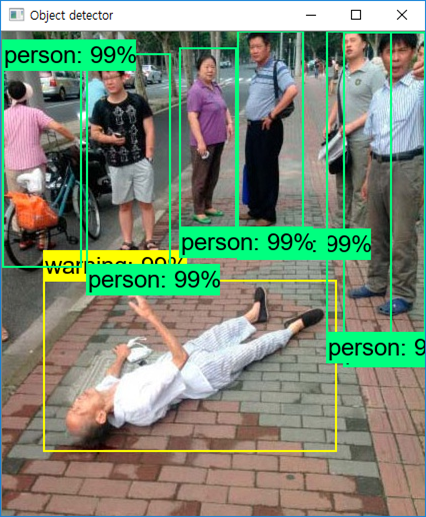
**3.1.2 머신러닝**

앞서 캡처한 이미지에 대한 객체를 감지하기 위해 기존 model인 Google社의 Inception V2 COCO Model에 추가 학습을 진행하였다. 쓰러진 사람과 보행자를 구별하기 위해, 쓰레기를 감지하기 위해 데이터를 구글 검색과 개인 촬영을 진행하여 1103장을 수집하였고, 수집한 각각의 이미지에 라벨링(이미지에 테두리를 씌워 객체 인식을 한 결과를 표시)을 하여 머신 러닝을 수행한다.

**[Figure3] 이미지에 대한 labeling**

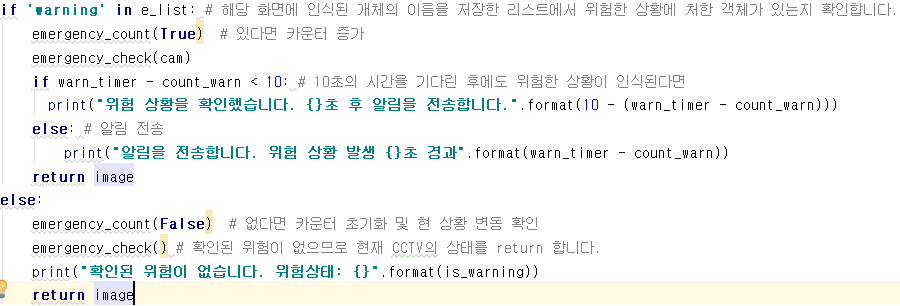
**[Figure4] 머신러닝으로 생성한 custom model의 object detection**

**3.1.3 인식종류**

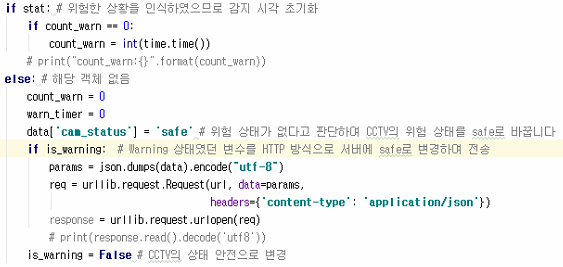
앞서 설명한 custom model을 사용하여 객체 인식을 수행하며, 객체를 인식할 때 사용하는 알고리즘을 서술한다. 추가 학습을 통해 구분한 객체의 종류는 다음과 같다.

* 쓰러진 사람

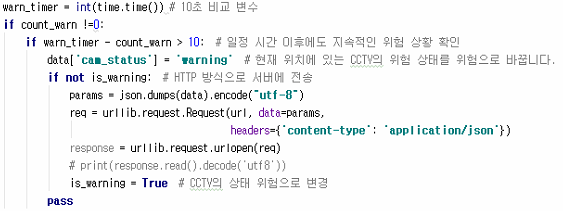
[Figure3,4]의 결과로 쓰러진 사람과 보행자의 구별이 무난히 수행되는 것을 알 수 있지만, 쓰러진 사람이 위급한 상황에 처했는지 알 수 없다. 따라서 쓰러진 사람을 판별하기 위해, 객체 인식을 통해 쓰러진 사람을 감지했을 경우, 10초의 시간을 대기하며 지속적으로 감지되는지 확인한다. 해당 객체가 의식이 있는 상태일 경우 몸을 움직이게 되어 객체 인식을 할 것이고, 의식이 없다면 움직임이 없으므로 지속적으로 감지가 될 것이다. 따라서 10초 동안 지속적으로 감지하여 쓰러진 사람이 있다고 판별할 경우, 해당 카메라의 dict 정보를 변경하고, 변경한 정보를 DB에 Update한다.



**[쓰러진 사람 판별 알고리즘]**



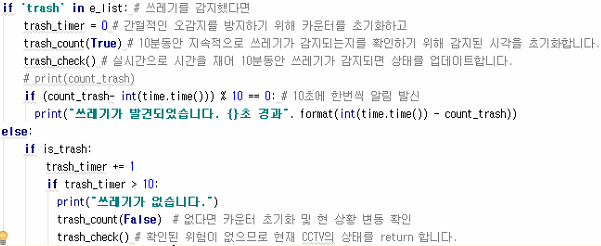
**[객체 감지 또는 미 감지 시 변수 초기화]**



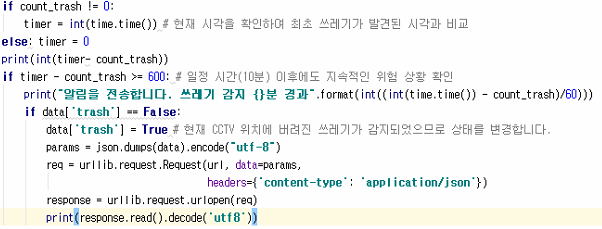
**[10초 대기 후 DB Update]**

* 쓰레기

쓰레기를 감지하는 즉시 알림을 전송하기에는 사람이 쓰레기를 들고 지나가는 경우가 있어 어려움이 따른다. 따라서 쓰레기가 버려진 후부터 판단을 시작한다. 먼저 쓰레기가 버려져 객체 인식을 통해 감지가 되었다면, 600초 동안 대기하며 쓰레기 투기 여부를 확인한다. 또한 10 프레임의 유예 기간을 두어 도중에 프레임을 인식하지 못해 다시 타이머를 초기화하는 경우를 대비한다.

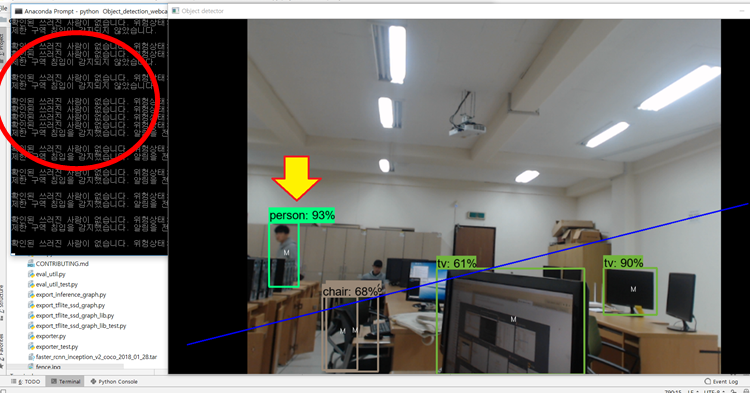


**[쓰레기 감지 알고리즘]**



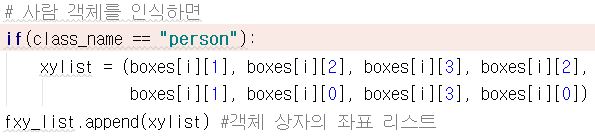
**[객체 감지 또는 미 감지 시 변수 초기화 | 매분 알림]**

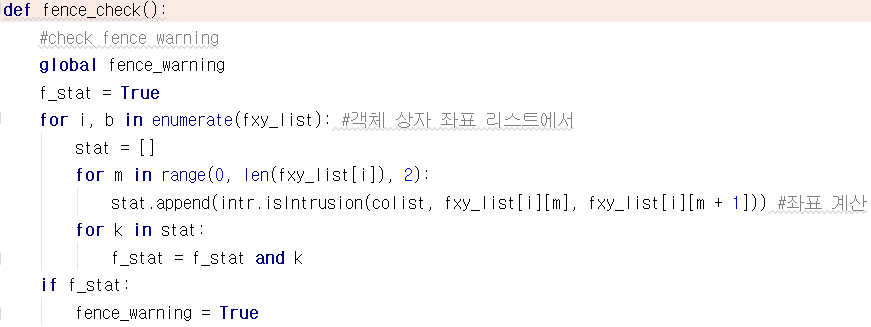
* 접근제한구역 침입자



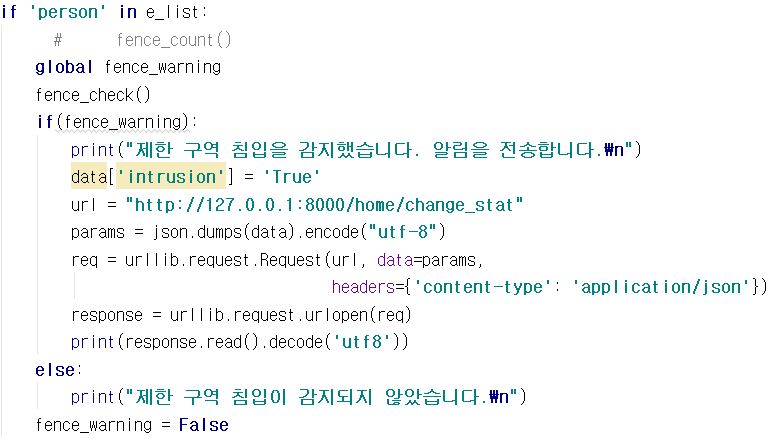
**[Figure5] 머신러닝으로 수행하여 감지한 가상펜스 침입**

접근 제한 구역에서의 가상 펜스는 OpenCV를 통해 제한 구역의 두 끝 점의 좌표와 OpenCV의 함수 cv2.line()을 통해 선을 그어 설정할 수 있다. 이후 사람 객체가 인식되었을 때 boundary box의 x, y 최대 최소 값을 받는다. 완전 접근 제한 구역에서의 침입을 감지하는 방법으로는 각 꼭지점 좌표를 (x, y) 튜플로 나타내고 각각의 x 값을 펜스 직선 방정식에 대입하여 모든 꼭지점의 초기 y 값이 그 결과값 y'보다 클 때, 즉 직선 위에 위치하고 있을 때 침입으로 감지하여 DB를 카메라 정보와 침입 감지 경고로 업데이트한다. 이 방법으로는 보행자와 침입자를 구분할 수 없는 구역의 경우에는 다른 방법을 적용한다. OpenCV의 MultiTracker\_create() 클래스로 다수 객체를 추적할 수 있는 클래스를 생성하고, 앞서 받아온 x, y 최대 최소 값으로 min x, min y, max x - min x, max y - min y 를 변수로 넣어 추적할 상자를 설정한다. tracker type이 CSRT인 tracker를 생성하여 30초 동안 각 객체의 움직임을 추적하고, 30초가 지나면 tracker의 상자를 설정한 x, y 변수를 다시 받아와 tracker를 초기화 하고 바뀐 상자를 추적한다. 이때 초기화 주기인 30초 동안 tracker window(추적 중인 객체 상자)가 위로 움직여 y 좌표의 수직 값 변화량이 30 픽셀 이상일 경우 DB를 카메라 정보와 침입 감지 경고로 업데이트한다.





**[객체 상자의 좌표와 가상 펜스를 설정할 좌표를 받아 가상 펜스의 침입 여부를 판단]**



**[제한 구역 침입 감지 시의 DB 업데이트 및 Boolean 변수 초기화]**



**[침입 감지를 위한 네 개 꼭지점의 좌표 계산 함수]**

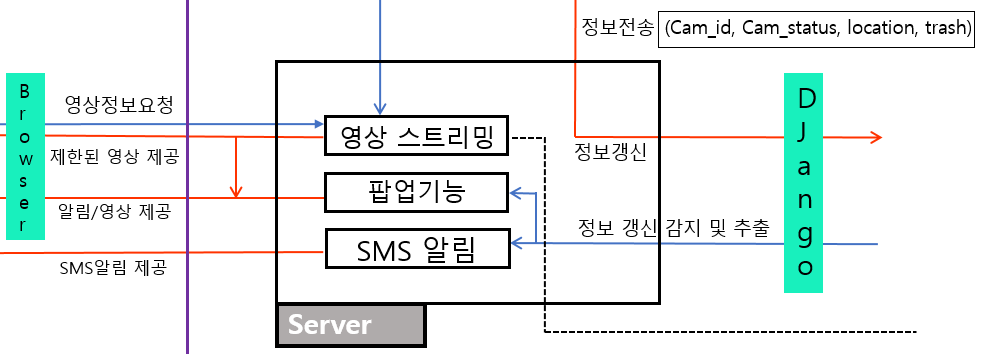
**3.1.4 인식결과**

1.3에서 기술한 3가지 인식 종류에 대한 객체 인식을 거치면 가공한 Frame을 return하며, 객체 인식한 결과를 토대로 선택적으로 DB를 Update한다.



**[카메라의 정보를 저장한 Dictionary 형태의 변수]**

**3.2 객체정보처리**



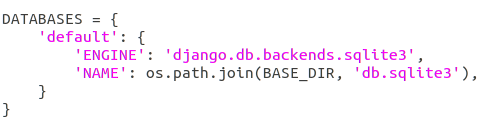
**[Figure6] 시스템 구성 요소2: 객체정보처리**

객체정보처리에는 정보수집, 알림 메시지 전송, 영상제공의 3가지 기능이 있다.

**3.2.1 정보수집**

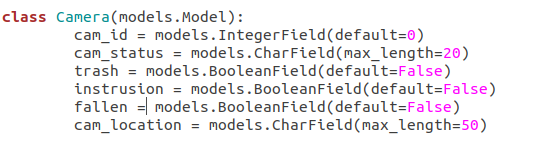
Object detection 시스템으로부터 받은 데이터를 수집하여, 데이터베이스에 이를 갱신하여 저장하는 기능이다. Models.py에서 Table을 생성하고 views.py에서 전송되는 데이터를 수신하며, 이를 데이터베이스에 저장하는 동작을 수행한다.

\*DB연결



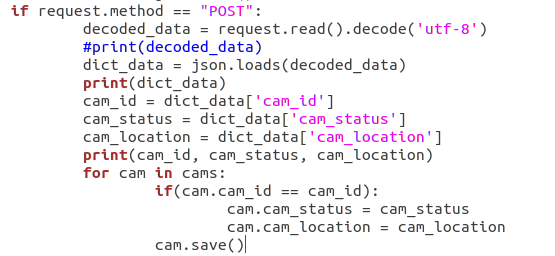
**[Django DB연결setting]**

\*DB table 생성 및 수정

****

**[정의된 DB Camera Table]**

models.py에서 class(table)와 field들의 attribute를 정의하고 python manage.py migrate를 통해 DB에 table을 생성 views.py에서 object detection 시스템으로부터 전송받은 데이터를 DB에 저장한다.



**[camera table의 field수정]**

**3.2.2 알림 메세지 전송**

DB의 갱신을 감지하여, 변경된 내용을 관리자에게 전송하고, 관리자이 변경된 사실을 알 수 있도록 알림을 제공한다. 이는 관리자의 정보요청에 의해 발생하는 것이 아닌, 서버 측에서 자동적으로 push해 주는 것이다. 또한 CoolSMS에서 제공하는 API를 사용하여 SMS 전송하여 이상 상황에 대한 카메라 정보, 경고 종류에 대한 정보를 관리자에게 발신하여 부재시에도 상황 파악 및 출동이 가능하다.

\*DB 갱신 감지

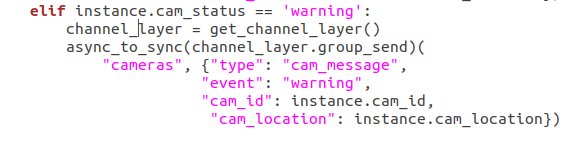


**[DB갱신감시 signal]**

Django에서는 특정한 일이 발생될 때마다 알려주는 signal을 발생하는 기능을 가지고 있다. 그 중 post\_save 는 DB model에 관련해서 save가 작동하면, signal을 발생시킨다.

\*알림 및 정보전송

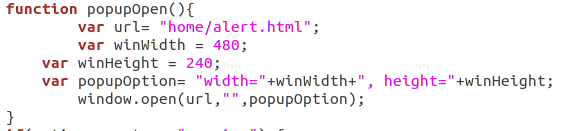
Signal이 발생하면 websocket을 통해 관리자의 browser에 실시간으로 변경된 정보를 push 하며, 자동으로 팝업 창을 띄워 그 안에 감지상황과 카메라의 번호, 위치와 찍히고 있는 영상을 제공한다



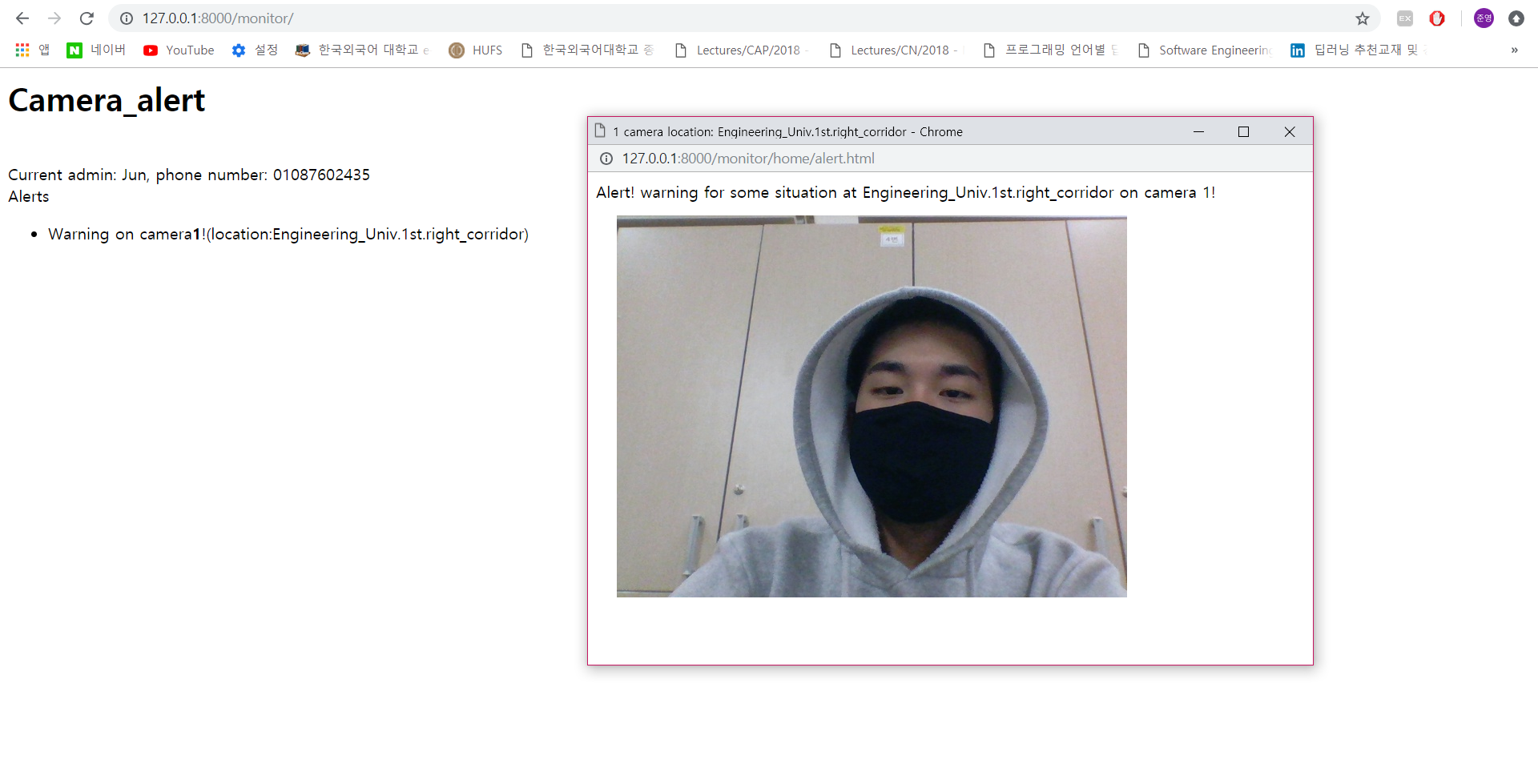
**[관리자의 Browser에 보내지는 Data값]**



**[관리자의 browser에서 실행되는 websocket(home.html)]**

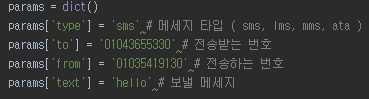
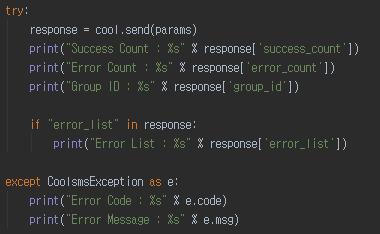


**[websocket안에서 실행되는 popupOpen함수]**

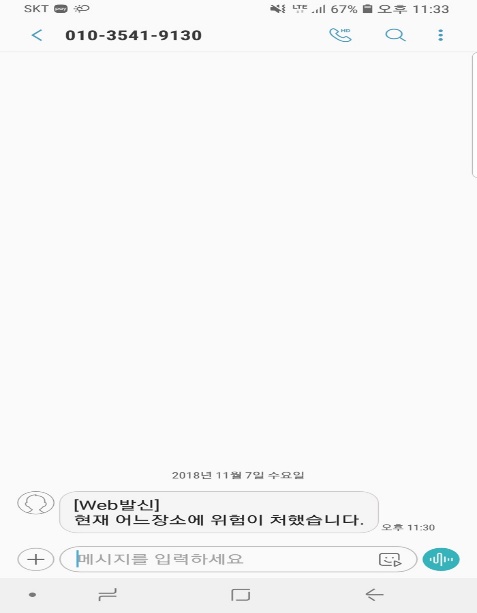


**[Figure7] 관리자 경고화면 팝업창**

\*SMS전송

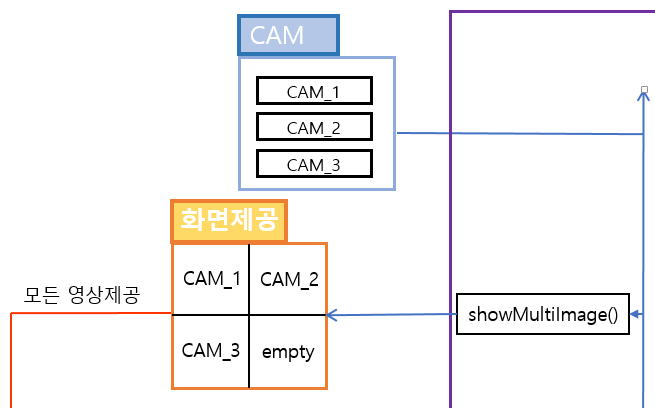


**[CoolSMS를 이용한 SMS전송]**

****

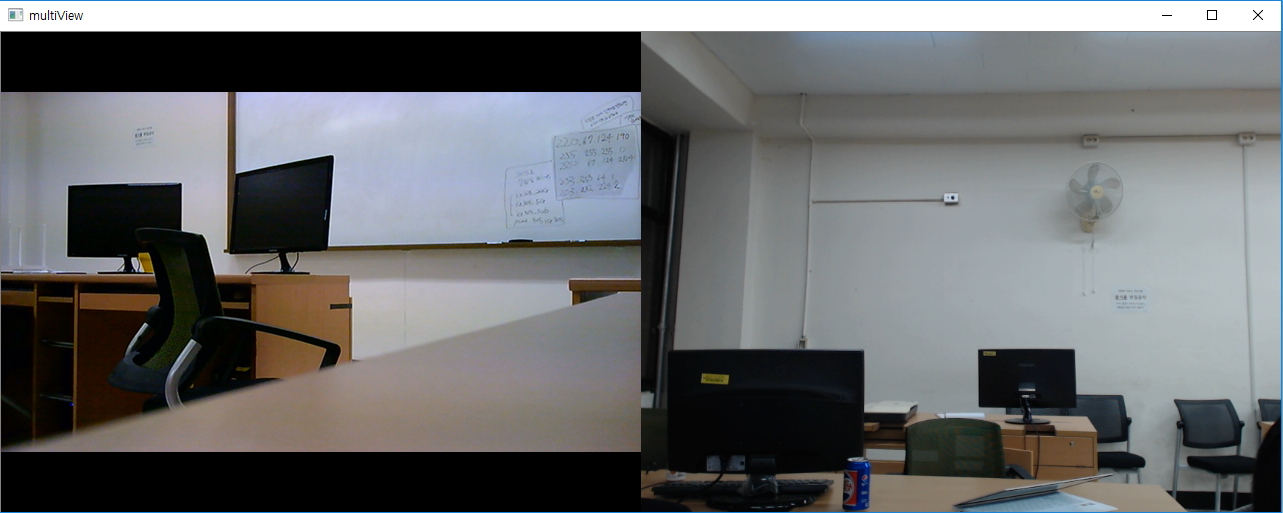
**[Figure8] SMS전송**

**3.2.3 영상제공**

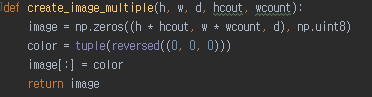


**[Figure9] 시스템 구성 요소3: 영상제공**

전체적인 영상을 관리자에게 제공하며 알림이 전송됨과 동시에 관리자이 확인할 수 있도록 영상을 실시간으로 제공한다. 또한 주민들도 원한다면 웹서버에 접속하여 실시간으로 제공되는 영상을 확인할 수 있다. 알림이 전송됨과 동시에 관리자가 확인할 수 있도록 알림 팝업창에 영상을 실시간으로 제공한다. 또한 주민들도 원한다면 웹서버에 접속하여 관리실에서 제공하는 영상들을 실시간으로 확인할 수 있다. OpenCV library 내부의 VideoCaptur함수를 이용하여 카메라의 영상을 캡쳐를 한다. 캡쳐한 JPEG frame들은 StreamingHttpResponse함수를 이용하여 스트리밍하여 영상을 보여준다.



**[Figure10] 전체적인 영상제공**





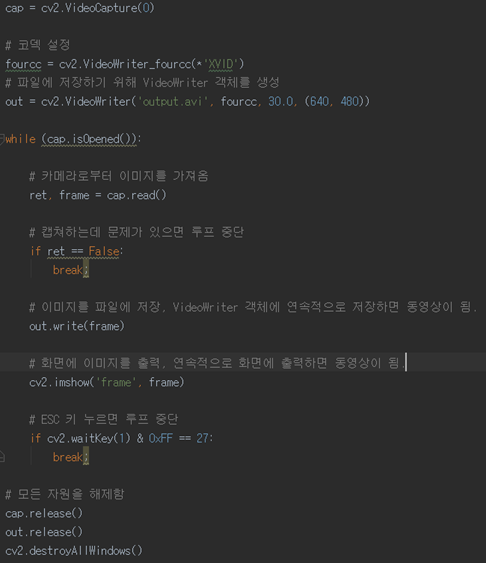
**[화면을 분할 showMultiImage()]**



**[영상 스트리밍(views.py)]**

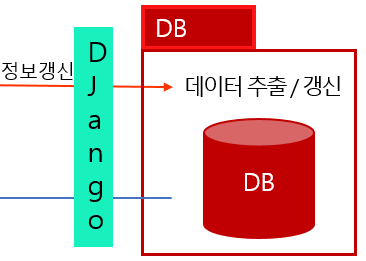
**3.2.4 영상저장**

OpenCV library 내부의 VideoWriter함수를 이용하여 영상 저장한다. 영상은 전부 녹화되어 저장되며 한달간 보관하게 된다.



**[영상저장 Code]**

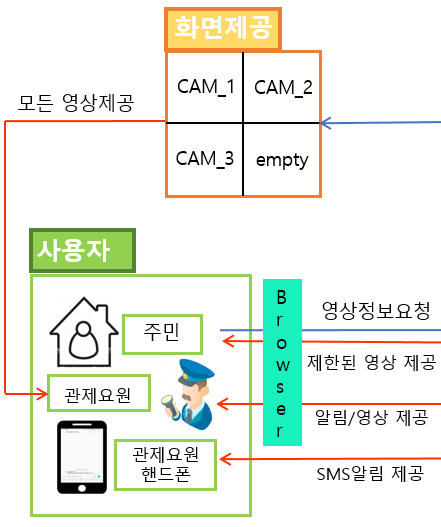
**3.3 DataBase**



**[figure11] 시스템 구성 요소4: DataBase**

주민과 관리자의 정보와 카메라 정보를 저장하기 위해 DataBase가 필요하다. 먼저, 주민과 관리자의 정보와 수집된 객체의 정보를 저장하기 위한 테이블을 생성한다. 이후, 객체인식으로부터 수집한 데이터를 데이터 처리 요청에 따라 저장하고 추출하는 기능을 한다. DataBase 동작은 다음과 같으며 상세 코드는 위에서 설명한 바와 같다.

**3.4 사용자**



**[Figure12] 시스템 구성 요소5: 사용자**

EDUM을 사용하는 주체로서 일반 주민과 관리자로 구분한다. 일반 주민은 서버에 데이터를 요청하여 제한되어 있는 영상을 하나를 제공받는다. 관리자는 자동으로 알림 메세지와 경고상황에 대한 영상을 제공받는다.

* + 1. **관리자**

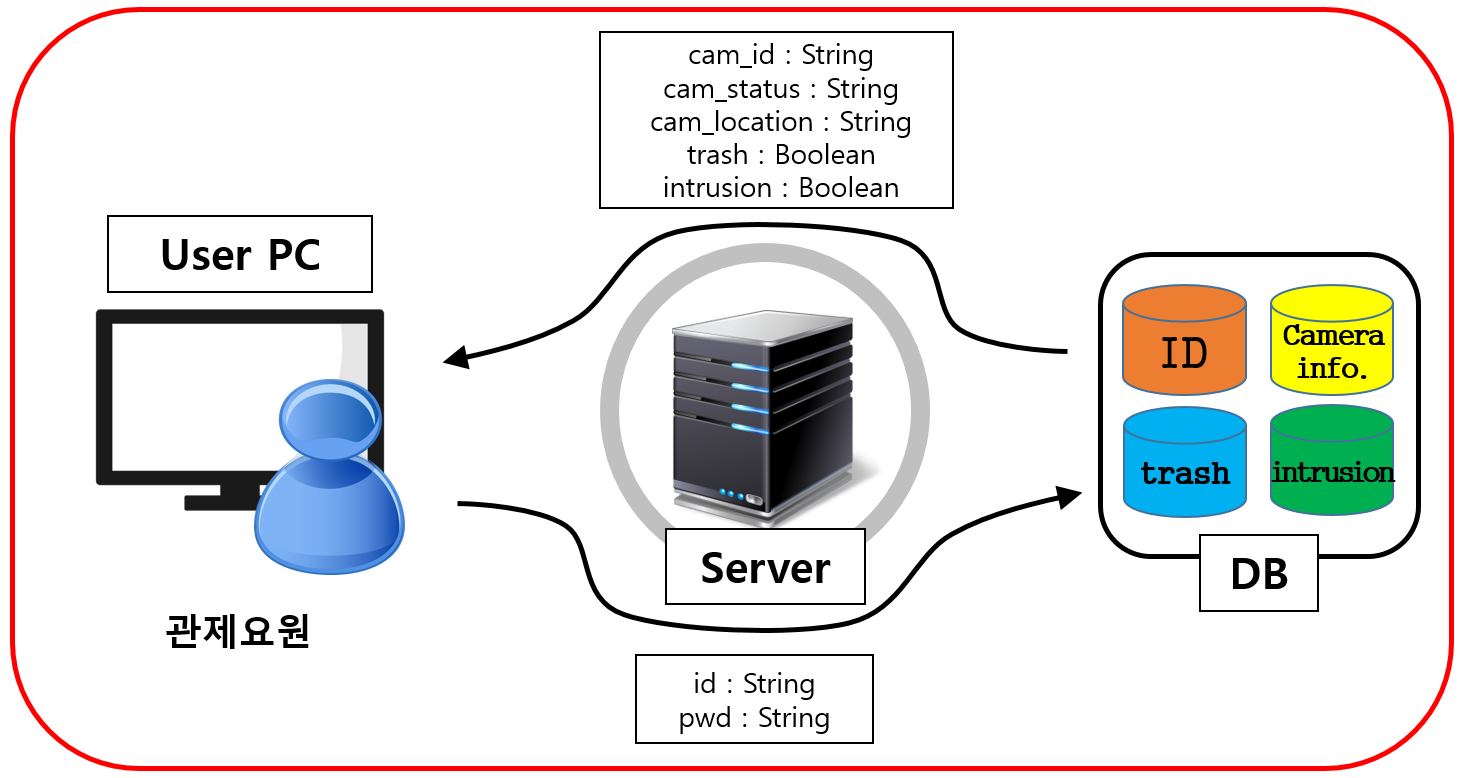
- cam\_id : 카메라 ID

- cam\_status : 쓰러진사람 유무

- cam\_location : 카메라 위치

- trash : 쓰레기 유무

- intrusion : 침입 유무



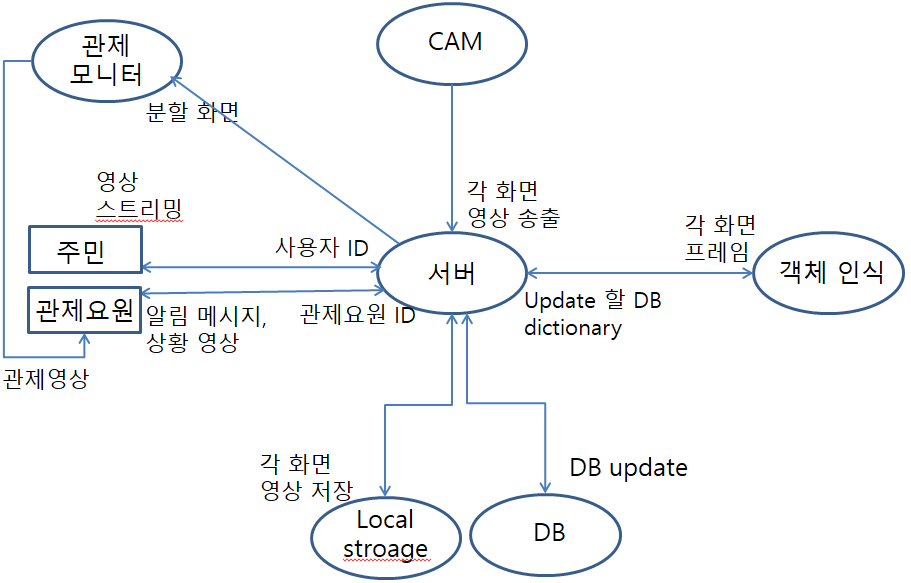
**[Figure13] 관리자의 Data요청 및 처리 과정**

**3.4.2 일반주민**

주민은 주민전용 웹페이지에 접속 후 로그인 시 관리자이 제공하는 제한된 영상들을 선택하여 실시간으로 볼 수 있다. 로그인 data요청 및 처리과정은 관리자의 data요청 처리와 같으며 다만 DB에서 받아오는 data는 없다.

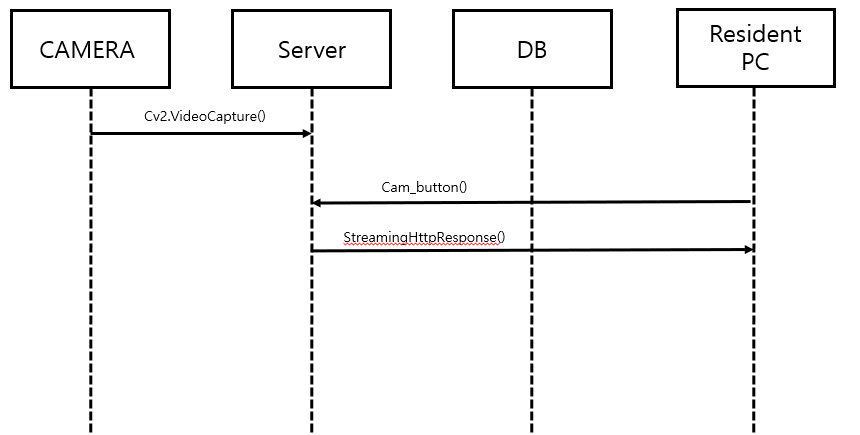
**4. 기능동작**

**4.1 Dataflow**

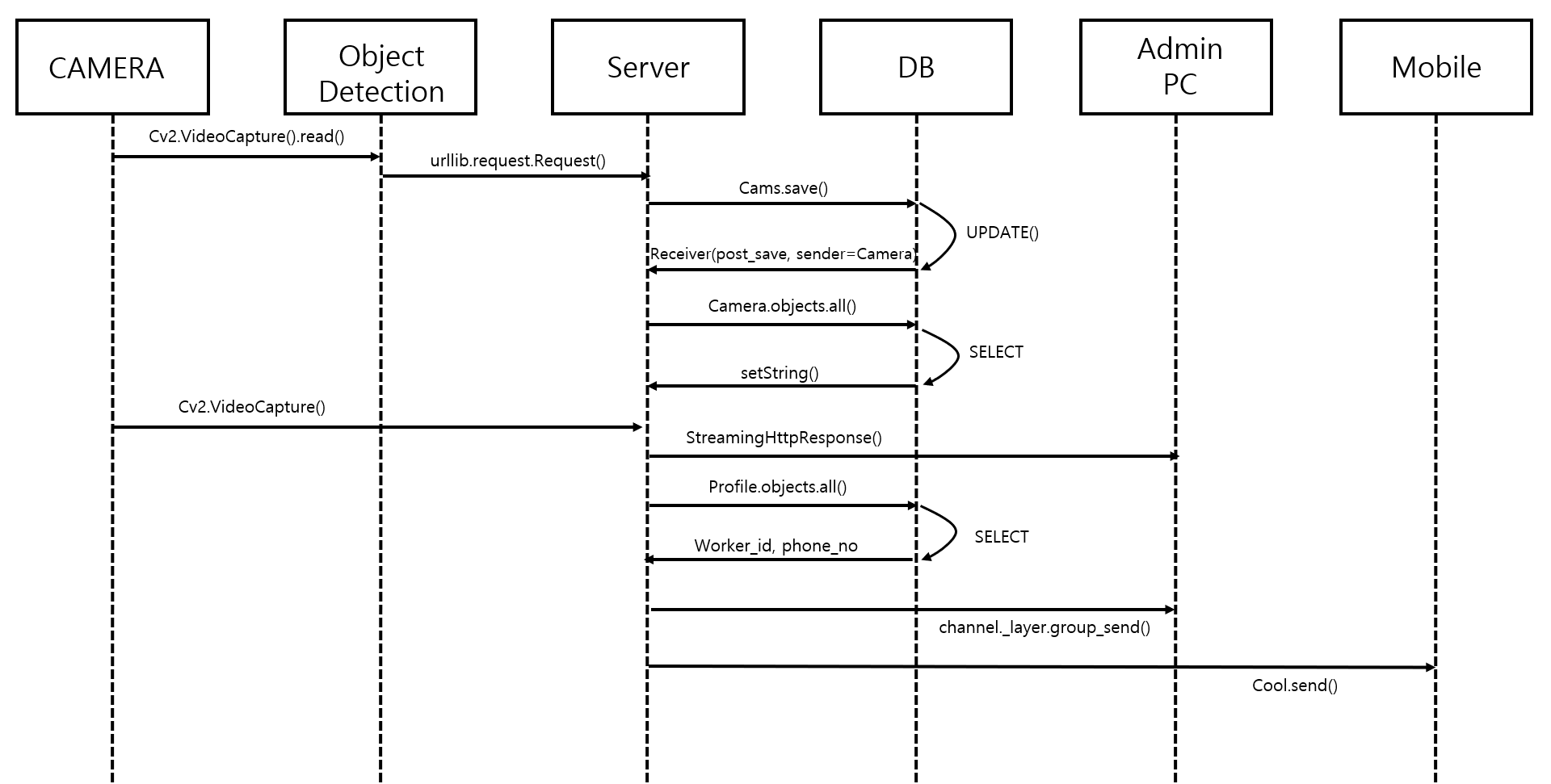


**[Figure14] Dataflow**

**4.2 Sequence Diagram**



**[Figure15] 관리자 입장의 Sequence Diagram**



**[Figure16] 거주민 입장의 Sequence Diagram**

**5. 자체 시험 방법 및 절차**

'EDUM'의 자체 시험은 영상의 흐름에 따라 단계별로 수행할 것이다. 각 단계별 시험 방법 및 절차는 아래와 같다.

* **영상수집**
  + 웹 카메라 두 대로 촬영되는 영상을 분할 화면을 통해 모니터와 서버로 송출하는 것을 확인
* **영상처리**
  + 입력 받은 영상을 5초마다 1프레임씩 받는지 확인
  + 프레임에서 3가지 이상 상황(쓰러진 사람, 접근제한 침입, 쓰레기 투기)이 모두 제대로 감지되는지 확인
  + 감지되었을 때 DB의 카메라 정보, 경고 종류가 업데이트 되는 지 확인
* **영상저장 및 스트리밍**
  + 영상이 로컬디스크에 압축되어 저장되는지 확인
  + 영상이 서버를 통해 스트리밍이 되는지 확인
* **영상이용**
  + 서버에서 데이터베이스로부터 정확한 데이터 추출확인
  + 추출한 데이터를 사용자에게 성공적으로 알람을 통해서 제공하는지 확인
  + 영상이 잘 나오는지 확인

**6. 향후 실제 적용방안**

* 상가단지, 다세대주택 혹은 아파트단지 등에서 사용 가능한 이 시스템은 주민의 편의와 안전을 위해 이용된다.
* 객체 인식을 통한 쓰러짐 감지를 통하여 인적 드문곳에서의 안전사고 등을 빠르게 대처 가능하다.
* 또한 쓰레기를 감지하여 무단 투기된 쓰레기가 있을 경우, 관리자에게 알려 해당 구역을 빠르게 청소할 수 있다.
* 제한구역을 설정하여 해당지역에 사람이 감지될 경우 즉시 관리자에게 알림을 줘서 통제 가능하도록 한다.
* 제한구역의 알림은 On/Off 컨트롤을 통해 알림 제공을 설정할 수 있다.
* 위 사항들을 통해 해당 구역 관리자의 관리능력이 저하되지 않고 지속적인 관리가 가능하도록 도와준다.

**7. 기대효과**



**[Figure17] 수동 영상관제 시간에 따른 위험 탐지율**

다음 그림에서 볼 수 있듯이 관리자들의 CCTV 감시 능력은 12분이 지나면 45% 가 떨어지고, 22분이 지나면 95% 가 감소한다. 따라서, 이로 인해 사고 발생에 대한 대처가 지연되거나 불가능할 수 있으며 관리자들에게 보다 편리하고 확실한 시스템을 제공할 필요가 있다.

현재 서비스되는 관리 시스템들은 실시간으로 영상 확인은 가능하지만 사건 발생시 사람이 직접 눈으로 확인해야 하며 만일 확인하지 못할 시 관리자는 이를 지각하지 못한 채 시간이 지나가 빠른 대처가 힘들다.

EDUM은 이러한 일을 방지할 수 있으며 관리자에게 보다 편리하고 확실한 서비스를 제공하여 관리 업무에 도움을 줄 수 있다. 사건발생 즉시 해당장소로 출동이 가능하며 관리의 효율이 높아진다.

**8. 세부 추진계획 및 일정**



**[Figure18] 추진계획 및 일정**