|  |
| --- |
| 윤 희영  2020-8-30 |

|  |
| --- |
| 프로젝트 보고서 |
| ELK를 활용한 CCTV 통합 관리 시스템 구축 |

목 차

[1. 개요 2](#_Toc50828290)

[1.1 프로젝트 추진 배경 2](#_Toc50828291)

[1.2 프로젝트 목표 2](#_Toc50828292)

[1.3 프로젝트 요약 2](#_Toc50828293)

[2. 프로젝트 구조 및 설명 3](#_Toc50828294)

[2.1 CCTV 3](#_Toc50828295)

[2.2 ALERT 5](#_Toc50828296)

[2.3 Elastic Stack 5](#_Toc50828297)

[3. 프로젝트 실행 및 결과 6](#_Toc50828298)

[3.1 개발 환경 6](#_Toc50828299)

[3.2 프로그램 실행 7](#_Toc50828300)

[3.3 소스 코드 10](#_Toc50828301)

[4. 참고자료 10](#_Toc50828302)

# 개요

## 프로젝트 추진 배경

집에서 사용할 가정용 보안 카메라를 알아보던 중 내가 원하는 기능만 넣어서 CCTV를 만들어 보고 싶다는 생각이 들었다. 집에 사용하지 않은 스마트폰이 있기 때문에 비용 면에서도 절약된다고 생각한다. 또한 최근에 Elastic Stack을 활용한 로그 관리 시스템에 대해 공부하고 있기에 CCTV와 접목시켜 프로젝트를 기획하였다.

## 프로젝트 목표

움직임을 감지하고 관리자에게 통보하여 즉각 대응하게 하는 CCTV 통합 관리 시스템을 구축하는 것이 목표이다.

## 프로젝트 요약

Python 언어와 데이터 분석 엔진인 Elastic Stack을 사용한다. 프로그램에는 ‘CCTV’와 ‘Alert’ 프로그램이 있다. CCTV 프로그램은 움직임을 감지하고 로그를 남기는 역할을 하며 해당 로그는 Logstash를 통해 구문분석 후 Elastic Search에 저장된다. Elastic Search에 로그가 저장이 되면 ‘Alert’ 프로그램에 의해 알람이 울리게 된다. ‘Alert’ 프로그램은 관리자용 프로그램이며 최근 로그들을 볼 수 있으며 수정도 가능하다. 이렇게 저장된 로그들은 Kibana 대시보드를 통해 종합적으로 확인 가능하다.

# 프로젝트 구조 및 설명

## CCTV

CCTV 프로그램은 움직임이 감지되면 로그가 저장되도록 설계되어 있다. 움직임 감지의 원리를 설명하기전에 동영상 재생에 대해 설명하자면, 동영상은 프레임(Frame)이 연속적으로 재생되어 보이는 것이다. 여기서 프레임(Frame)은 영상에서 보여지는 정지된 화면이다. 그래서 초당 보여주는 프레임이 많을수록 동영상이 부드럽게 재생된다. 이 프로그램에서 카메라 영상 처리를 위해 OpenCV 라이브러리를 사용했다.

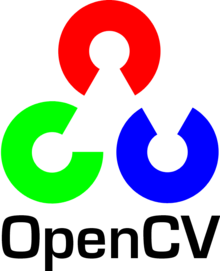


그림 - OpenCV 로고

OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 오픈소스 라이브러리이다. 안면인식, 물체추적 등 다양한 기술이 있지만, 이 프로젝트에서는 기본적인 영상 처리에 대한 기술만 사용한다.

OpenCV는 출력함수로 프레임을 출력해보면 다차원의 행렬이 나온다. [그림 2]를 보면 컬러의 경우 BGR 순서로 픽셀의 색상 값을 나타내고 있다. 반대로 흑백의 경우는 밝기의 정도로 0~255으로 표현되며 높을수록 밝다.

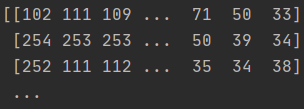
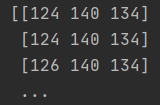


그림 - 컬러(좌) / 흑백(우)

움직임 감지에 있어서 컬러 정보는 불필요한 정보이므로 흑백으로 변경한다. 움직임 감지는 현재 프레임과 이전 프레임의 밝기 수치를 비교하는 방식이다. 우선 두 프레임의 픽셀의 밝기 수치를 뺀 후 제곱하여 음수 값을 없앤다. 그리고 프레임의 총 픽셀 수만큼 나누고 나눈 결과 값을 기준으로 하여 움직임을 판별한다. 프레임 간의 변화가 없으면 픽셀의 밝기 수치도 변하지 않기 때문에 결과값이 0에 가깝게 나온다. 반대로 변화가 있으면 변화의 정도에 따라 결과값이 높게 나오기 때문에 환경에 따라 적절한 제한 값을 설정하여 사용한다.

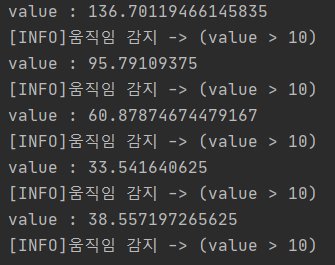


그림 - 움직임 감지

카메라를 작동시킬 때부터 상시 녹화가 되지만 모션이 감지가 되면 별개로 부분 녹화가 진행된다. 또한 Elastic Stack에 보낼 로그도 저장하는데 로그 내용은 [그림 4]와 같이 CCTV위치, 위도, 경도, 날짜들이 공백으로 구분되어 있다. 이는 Logstash에서 구문 분석을 위해 구분해 놓은 것이다.

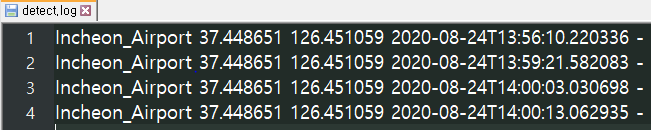


그림 - 로그 파일

## ALERT

Alert 프로그램은 CCTV에서 움직임이 발생하였을 때 관리자에게 팝업을 생성하는 알림 프로그램이다. 최근 로그와 이전 로그를 비교해서 다르면 팝업창이 뜨게 설정했다. 로그를 불러오는 작업은 Elastic Search 라이브러리를 사용했고, 미리 입력시킨 쿼리문을 날려 검색한다. 알림 외에 감지된 오브젝트 정보에 대한 추가 및 수정도 가능하다. CCTV 확인 결과 아무 이유없이 알림이 발생할 수도 있고 인가된 관리자 또는 신원미상의 출입자 등 알림 원인에 대한 내용을 표시할 수 있다.

## Elastic Stack

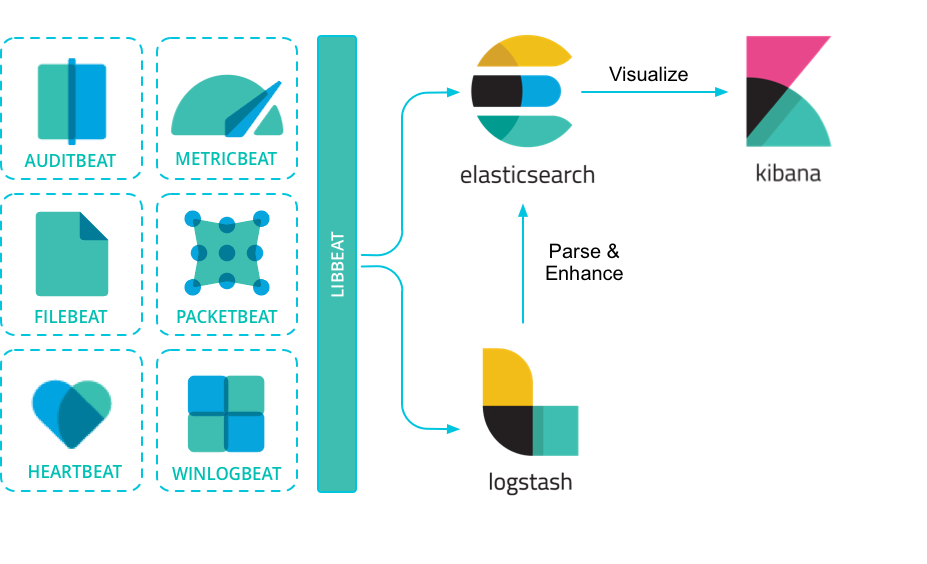


그림 – 출처 : elastic.co

Elastic Stack은 [그림 5]와 같이 Elastic Search, Logstash, Kibana 그리고 Beat로 이루어져 있다. Beat의 종류는 여러가지가 있는데 로그를 전송해주는 Filebeat를 사용한다. Filebeat의 설정은 Logstash의 아이피, 포트 그리고 전송할 파일 경로만 설정하면 로그 파일에 내용이 새로 입력될 때마다 Logstash로 전송된다. 로그 파일의 내용[그림 4]을 살펴보면 위치, 위도, 경도, 날짜/시간 순서로 되어있다. 이제 ‘logstash.conf’ 파일을 통해 해당 로그를 어떻게 가공할지를 작성해야 한다. 설정파일은 input, filter, output으로 나누어져 있다. input은 어떻게 입력 받을지 정하는 곳이다. Logstash로 직접 파일 경로를 입력할 수 있지만 Filebeat를 사용하므로 포트만 설정해 주면 된다. Filter는 데이터 처리 방식을 작성하는 곳이다. 여기서 가장 중요한 부분이 위도, 경도의 타입을 변환해줘야 한다. 위도, 경도는 소수점으로 이루어져 있어 ‘float’ 타입으로 변경한다. 또한 Elastic Search에서 위도, 경도의 타입을 ‘geo\_point’로 미리 매핑해야 하며, 이는 CCTV의 위치를 지도에 나타낼 때 사용된다. 마지막으로 output은 filter를 걸쳐 가공된 데이터를 어디로 보낼지 작성하는 곳이다. Elastic Search의 아이피와 포트를 적으면 되고 인덱스 이름과 보안이 걸려 있다면 아이디/패스워드를 이곳에 작성하면 된다. 이제 데이터를 가공해서 저장까지 했다면 그 저장된 데이터를 시각화해주는 것이 Kibana이다. Kibana는 대시보드를 제공하는데 예를 들어 탐지 위치 분포도, 탐지 원인 분포도, 탐지된 CCTV 지도 표시 등을 구성하여 현재 상황을 쉽게 파악할 수 있고 실시간으로 대응이 가능하다.

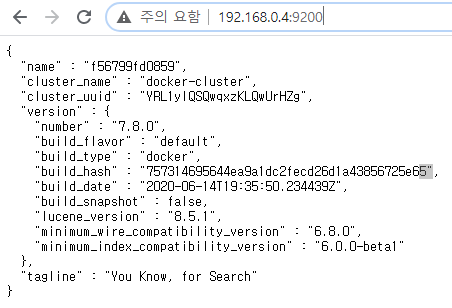
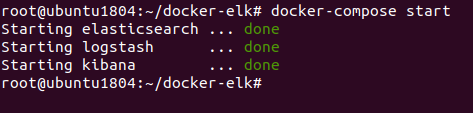
# 프로젝트 실행 및 결과

## 개발 환경

|  |  |
| --- | --- |
| 구분 | 시스템 |
| 하드웨어 | 스마트폰(카메라 어플 사용) |
| 운영체제/소프트웨어 | Window 10 - Anaconda3  Ubuntu 18.04 - Docker 19.03.12 |
| 언어/프레임워크 | Python 3.7  Elastic stack 7.8.0 |

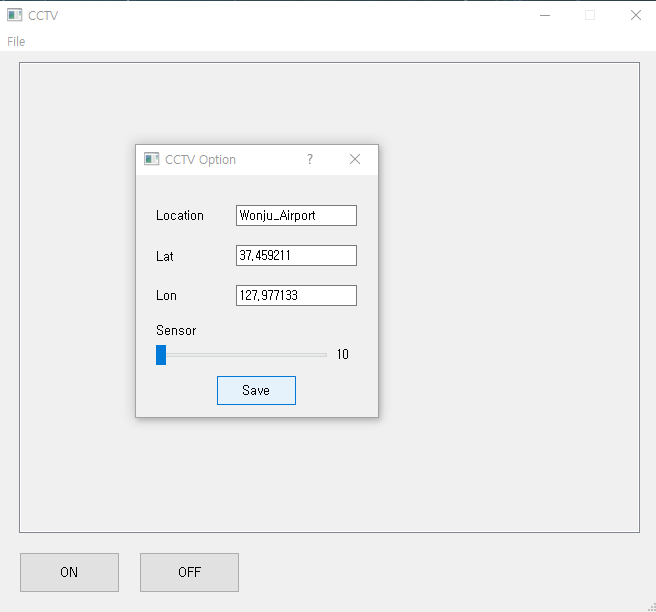
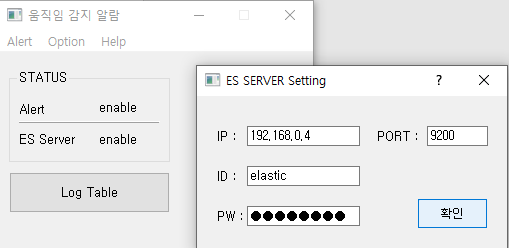
CCTV와 ALERT프로그램은 윈도우 운영체제에서 파이썬 언어로 개발하였고 CCTV 카메라가 없어 스마트폰으로 대체했다. ELK는 설치와 설정이 간편한 Docker 환경에서 진행했다.

## 프로그램 실행



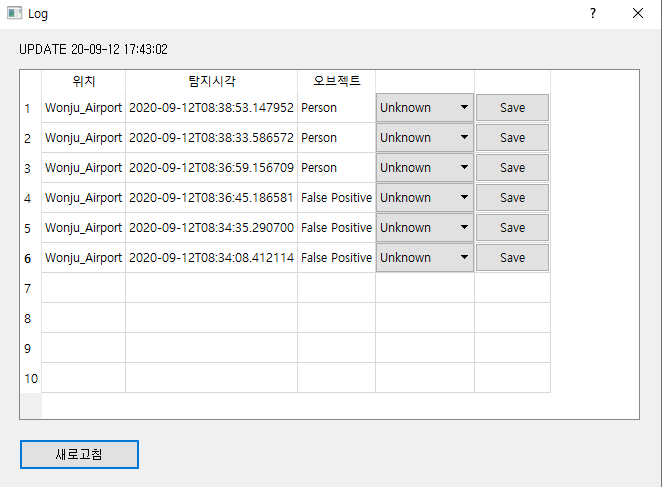
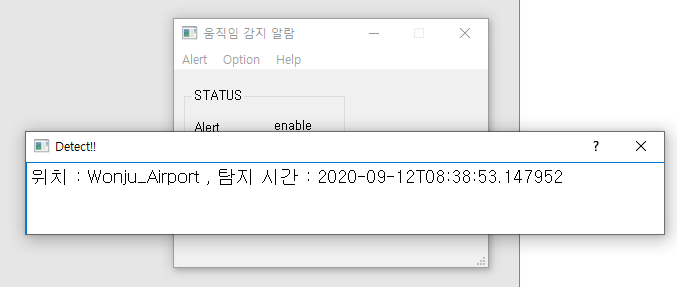
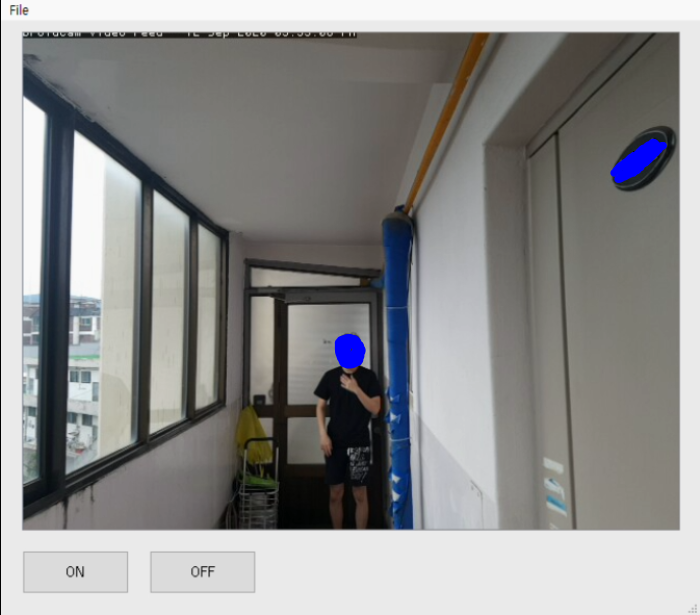
그림

[그림 6] Docker로 Elk 이미지를 올리고 실행시킨 후 서버 상태를 확인한다.

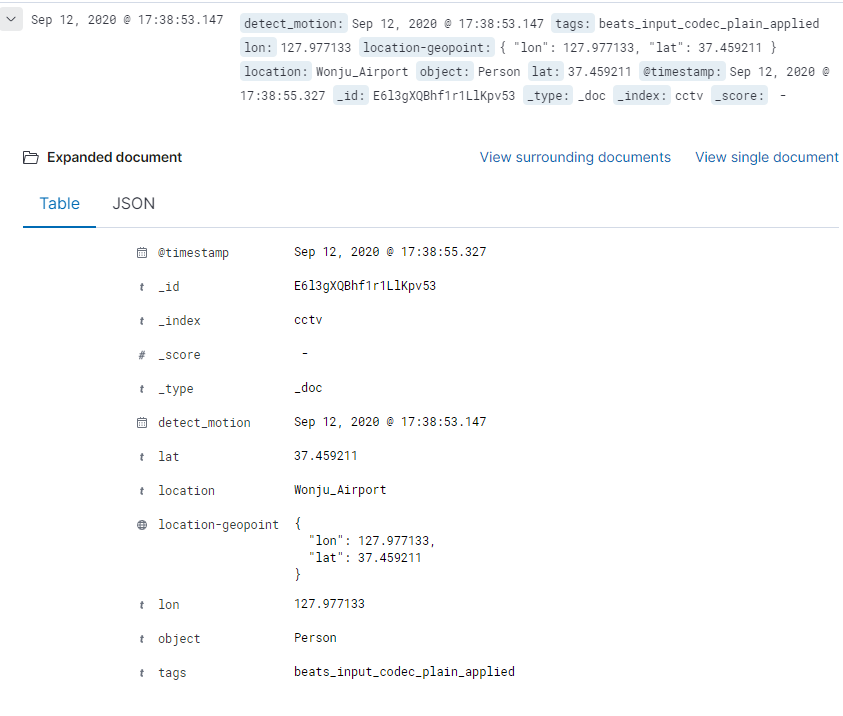
그림

[그림 7] CCTV 프로그램을 실행하여 위치 정보와 탐지 감도를 설정하고 카메라를 작동시키고, ALERT 프로그램도 ES의 정보를 입력 후 정상 작동하는지 확인한다



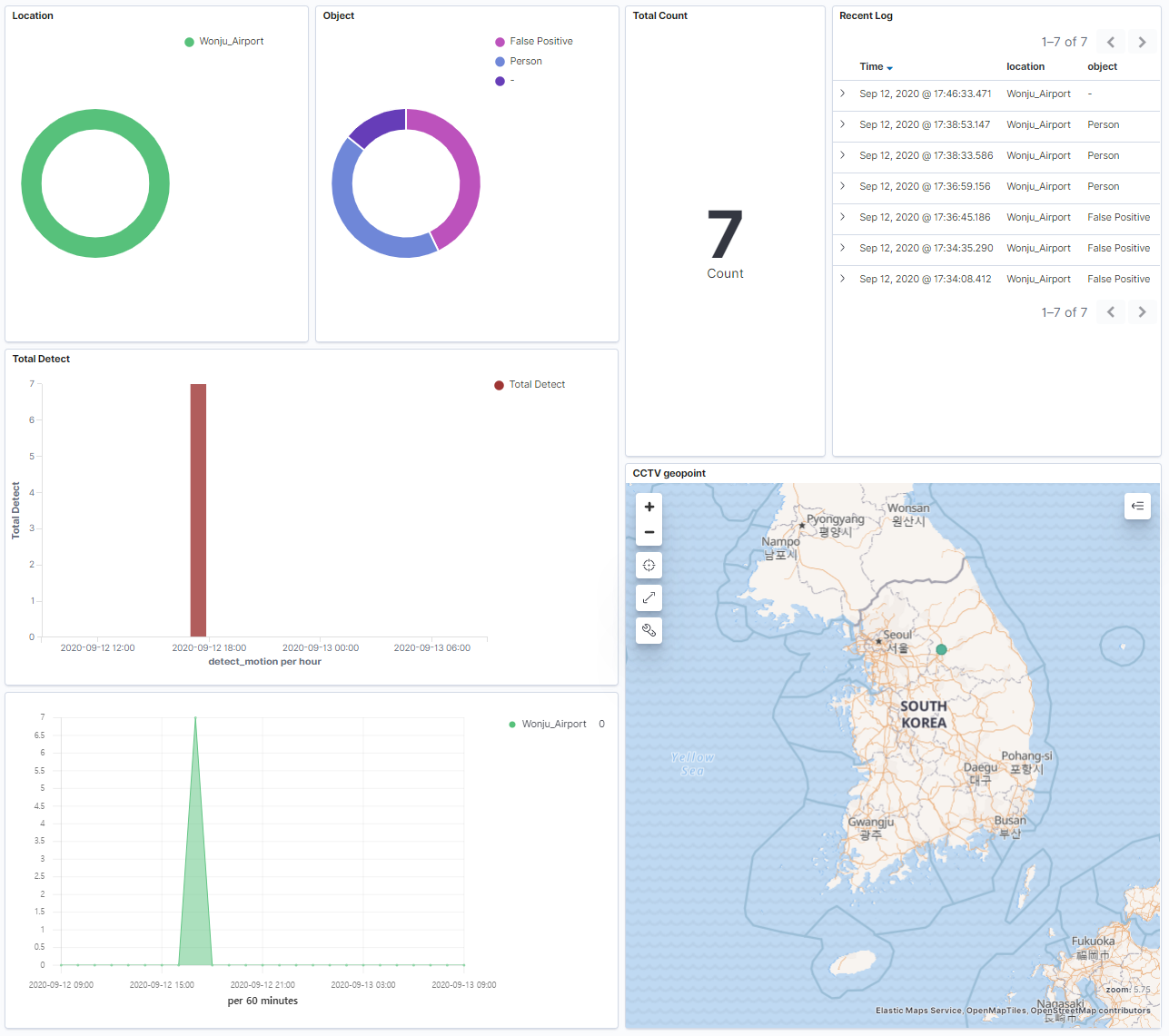
그림

[그림 8] 카메라 장소가 원주 공항의 출입 금지 장소라 가정한다. 사람이나 어떠한 움직임이 감지되었을 때 팝업이 발생하는 것을 볼 수 있다. 또한 CCTV 녹화본을 확인하여 감지된 오브젝트의 종류를 추가할 수 있다.



그림

[그림 9] 로그를 살펴보면 Logstash로인해 내용에 맞게 필드에 입력된 것을 확인할 수 있다. 이 로그를 기반하여 대시보드를 작성해 [그림 10]처럼 종합적으로 확인이 가능하다.



그림

## 소스 코드

<https://github.com/heisyoung/ELK-Project/tree/master/CCTV>

# 참고자료

|  |  |
| --- | --- |
| 구분 | 웹 사이트 |
| OpenCV | <https://opencv.org/> - |
| Elastic Stack | <https://www.elastic.co/>  <https://elasticsearch-py.readthedocs.io/en/master/index.html> |
| Docker | <https://github.com/docker/compose/releases>  <https://github.com/deviantony/docker-elk> |
| Python / PYQT5 | <https://wikidocs.net/1418>  <https://wikidocs.net/35478> |