

안티드론조

# 사이버보안 캡스톤디자인

중간평가 발표





# 사이버보안 캡스톤디자인

중간발표

## CONTENTS

01

### 프로젝트 목적

프로젝트의 궁극적 목표를 설명한다

02

### 선행연구

프로젝트를 시작하기전 필요한 지식

03

### 프로젝트 개발내용

무엇을 개발하였는가

04

### 향후 계획

중간발표 이후 계획을 설명한다

05

### 결과 및 토의

중간발표까지의 결과를 설명하고 토의해본다

06

### 참고 문헌

프로젝트를 진행하면서 도움을 받은 자료 공유

## 00 드론의 위협



2019년 9월 이란 드론 테러



2018년 8월 베네수엘라 대통령 연설



2018년 12월 게트윅공항 마비

## 01 프로젝트 목적



Step. 1

▶ 드론 사업 지원



협

기술 및 시스템 필요

위협이 되는 드론을 잡아라!!



## 02 선행 연구

---

1. Opencv
2. 객체 인식 방법 – yolo, ssd, r-cnn
3. flask



## 02 선행 연구 – Opencv

---

1. 인텔이 개발
2. OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리  
(다양한 영상/동영상 처리에 사용할 수 있는 오픈소스 라이브러리)
3. 실시간 이미지 프로세싱에 중점을 둔 라이브러리.
4. OpenCV는 TensorFlow , Torch / PyTorch 및 Caffe의 딥러닝 프레임워크를 지원한다.
5. OpenCV는 C/C++ 프로그래밍 언어로 개발 되었으며 Python , JAVA 및 MATLAB / OCTAVE에 바인딩 되어 프로그래머에게 개발 환경을 지원

## 02 선행 연구 – Opencv

그래서 Opencv로 무엇을 할 수 있는것인가?

동영상 사진 불러오기 후처리 이미지 윤곽선 검출 노이즈제거 얼굴인식 등 무궁무진





## 02 선행 연구 – DeepLearning 객체 인식/탐지

---

### 딥러닝

인공신경망 기반 (ANN: artificial neural network)의 기계학습 기술

- 인간의 신경망 구조를 모사하여 만든 인공 신경망을 기반으로 여러 데이터를 이용하여 사람처럼 스스로 학습하는 기계 학습 기술

깊은 레이어 구조를 갖는 심층 신경망(DNN: Deep Neural Network)

- 머신러닝과 다르게 Input layer, Hidden layer, Output layer 등의 여러 개의 layer를 갖는 심층 신경망 구조로 데이터가 많아 질 수록 높은 정확도를 보임



## 02 선행 연구 – DeepLearning 객체 인식/탐지

딥러닝을 가능하게 하는 요소들

인공 신경망 알고리즘

- 인간의 두뇌를 모사한 인공지능 알고리즘 등장
- 다양한 종류의 인공 신경망 모델 개발

Computing Power

- 병렬 연산이 가능한 GPU가 방대한 데이터 연산을 수행하는데 적합 함 인증
- 2015년 이후 GPU의 도입으로 딥러닝 발전 가속화

Big Data

- 방대해 지는 데이터
- 학습 데이터량 증가

개발 환경의 진화

- 인식 가능한 객체 종류 수 증가
- 인식률 향상



## 02 선행 연구 – DeepLearning 객체 인식/탐지

---

딥러닝 모델

인공신경망(ANN)을 기반으로 구성

신경망의 다양한 형태와 전처리 등을 조합한 모델 설계

주요 모델

CNN, RCNN RNN GAN DQN

딥러닝 플랫폼

TensorFlow

Caffe

Torch

CNTK

Theano



## 02 선행 연구 – DeepLearning 객체 인식/탐지

---

객체 인식/탐지 모델

R-CNN(Regions with CNN)

- 관심 영역(ROI) 추출 알고리즘과 CNN이 결합된 신경망 모델
  - 이미지 안의 객체들을 추적, 어떤 물체들이 있는지 추적
  - 입력 이미지에서 수많은 Object 후보들을 찾아내고 이들을 모두 CNN에 넣어서 특징을 추출
- 적용 분야
  - 이미지 안에 있는 객체 검출, 신호등 판별, 보행자 감지등



## 02 선행 연구 – DeepLearning 객체 인식/탐지

### 객체 인식/탐지 모델

#### YOLO(You Only Look Once)

- 이미지를 영역으로 분할하고 각 영역의 Bounding Box와 확률을 예측하는 하나의 신경망을 가진 모델
  - 이미지를  $S \times S$ 개의 그리드로 분할하고 그리드의 신뢰도를 계산
  - 신뢰도를 계산하여 Bounding Box의 위치를 조정함으로써, 가장 높은 객체 인식, 정확성을 가지는 Bounding Box를 추출
  - 이미지를 한 번에 바라보는 방식으로 클래스를 분별하여 처리속도가 빠름
- 적용 분야
  - 다양한 객체 검출



## 02 선행 연구 – DeepLearning 객체 인식/탐지

---

객체 인식/탐지 모델

SSD(Single Shot Detector)

- 객체 검출 속도 및 정확도 사이의 균형이 있는 알고리즘
  - 이미지에 대한 CNN을 실행한 후 형상 맵(Feature map)을 계산
  - Bounding Box 및 객체 분류 확률을 예측하기 위해 형상 맵을 CNN 수행
  - CNN 처리 후 경계 상자를 예측, 다양한 스케일의 물체를 검출
- 적용 분야
  - 다양한 객체 검출

## 02 선행 연구 - flask



플라스크는 특별한 도구나 라이브러리가 필요 없기 때문에 마이크로 프레임워크라 부른다.

데이터베이스 추상화 계층, 양식 유효성 확인, 기타 기존의 서드파티 라이브러리가 공통 기능을 제공하는 구성 요소가 없다.

그러나 플라스크는 플라스 자체에서 구현된 것처럼 애플리케이션 기능을 추가할 수 있는 확장 기능을 지원한다. 확장 기능은 객체 관계 매퍼, 양식 유효성 확인, 업로드 관리, 다양한 개방형 인증 기술, 여러 공통 프레임워크 관련 도구들을 위해 존재한다.

확장 기능들은 코어 플라스크 프로그램에 비해 훨씬 더 정기적으로 업데이트된다.

## 프로젝트 요소



**A.I**

DeepLearning – SSD를  
이용한 Object Detect



**ROBOT**

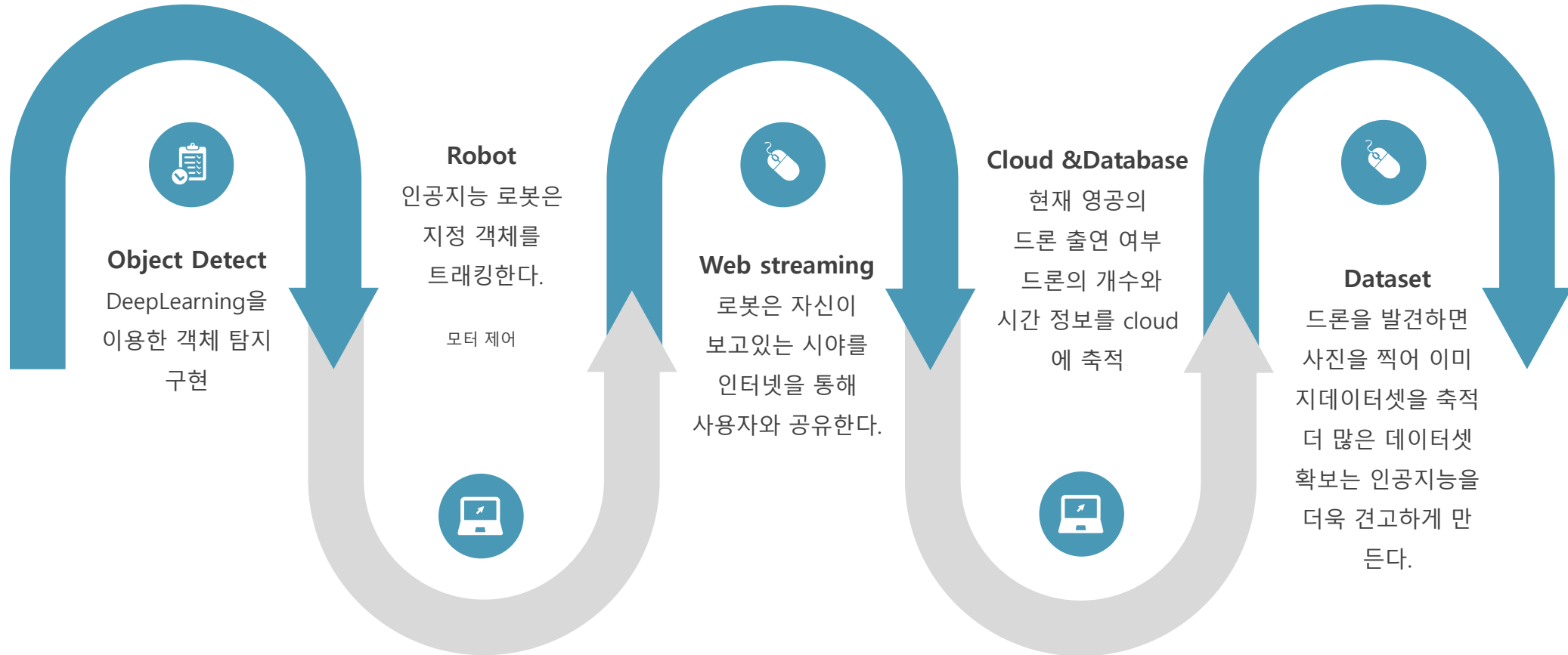
Object Tracking을  
하기위한 2 자유도 로봇



**IoT**

인터넷을 통해 로봇과  
사용자간의 실시간 정보공유

## 03 프로젝트 개발내용





# 04 DISCUSS FUTURE PLANS

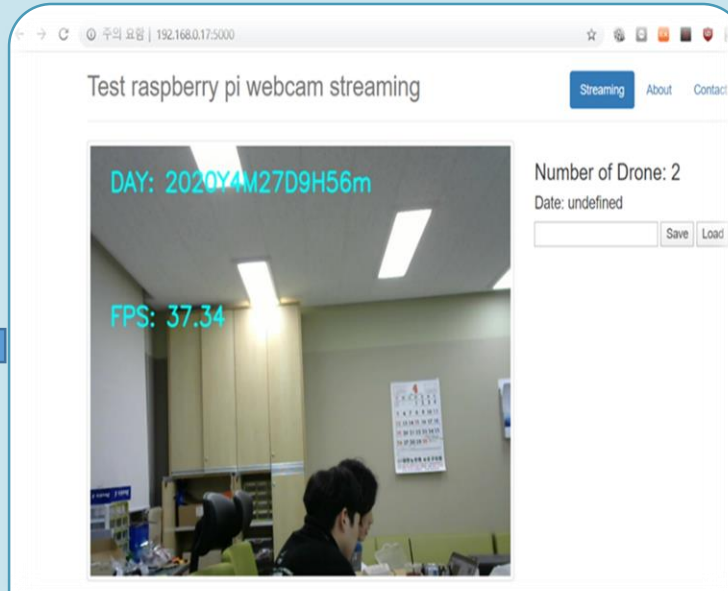
향후 계획



Object Detecting

## 드론 탐지

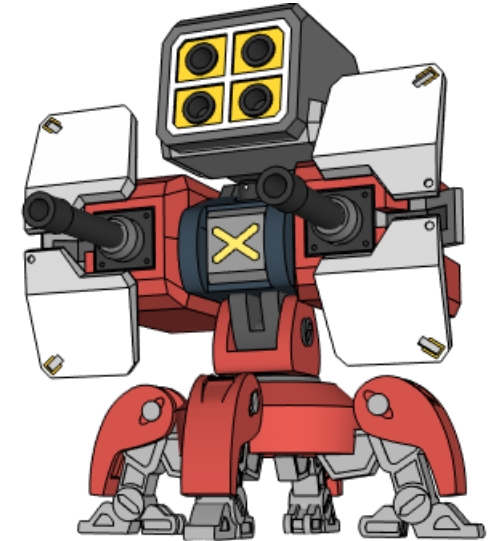
드론을 탐지하고 바운딩박스가 중앙에 있게 유지  
스텝모터2개를 이용



Web Monitoring

## 모니터링

웹에서 실시간 정보 모니터링  
라즈베리파이에서 얻는 모든 정보를 웹을  
통해 실시간 모니터링



최종 목표

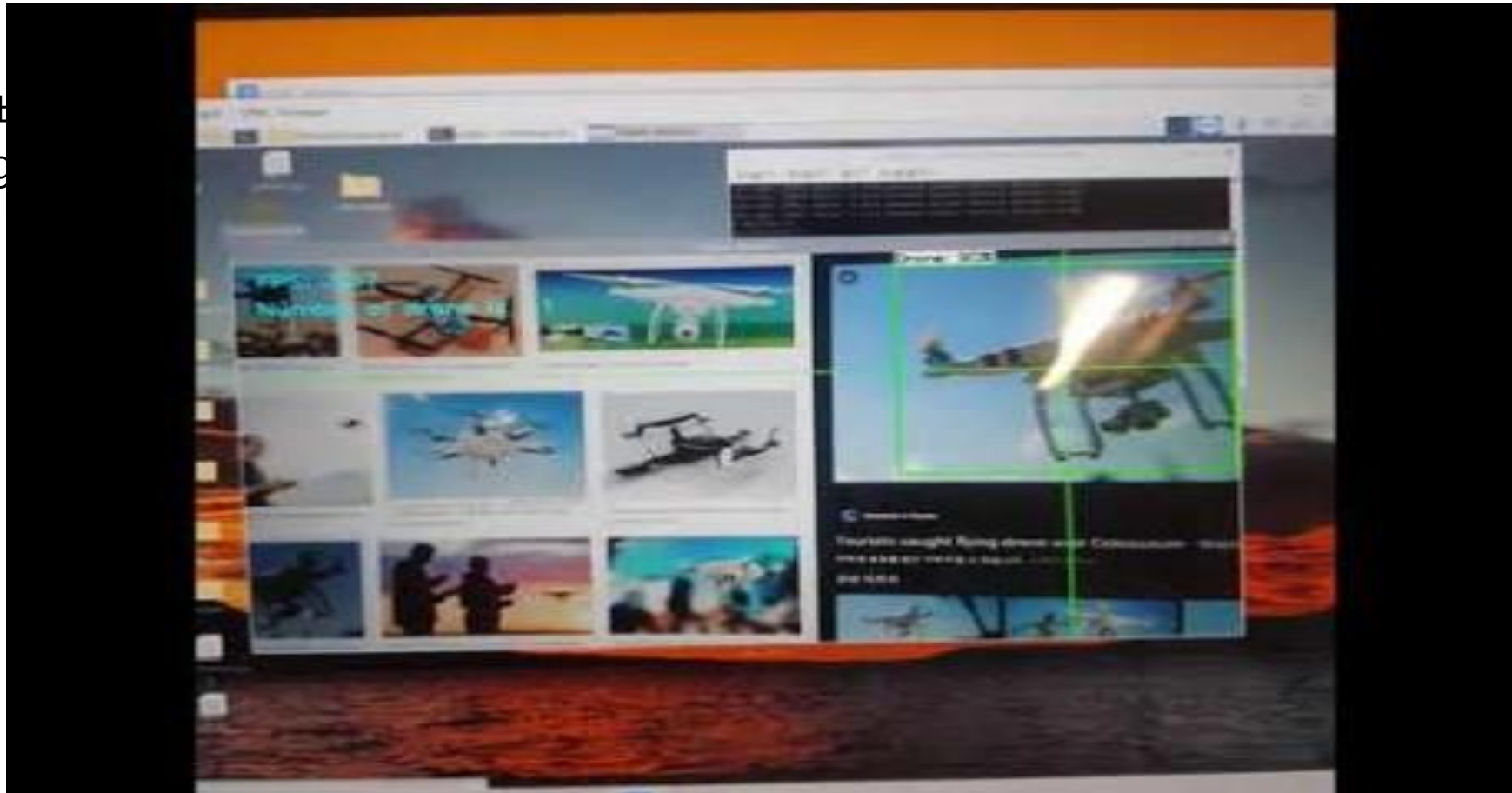
## A.I IoT Robot

지금까지 과정을 합치고 그 기능이  
탑재된 로봇

## 05 결과 및 토의

결과 리뷰 – ssd 선택

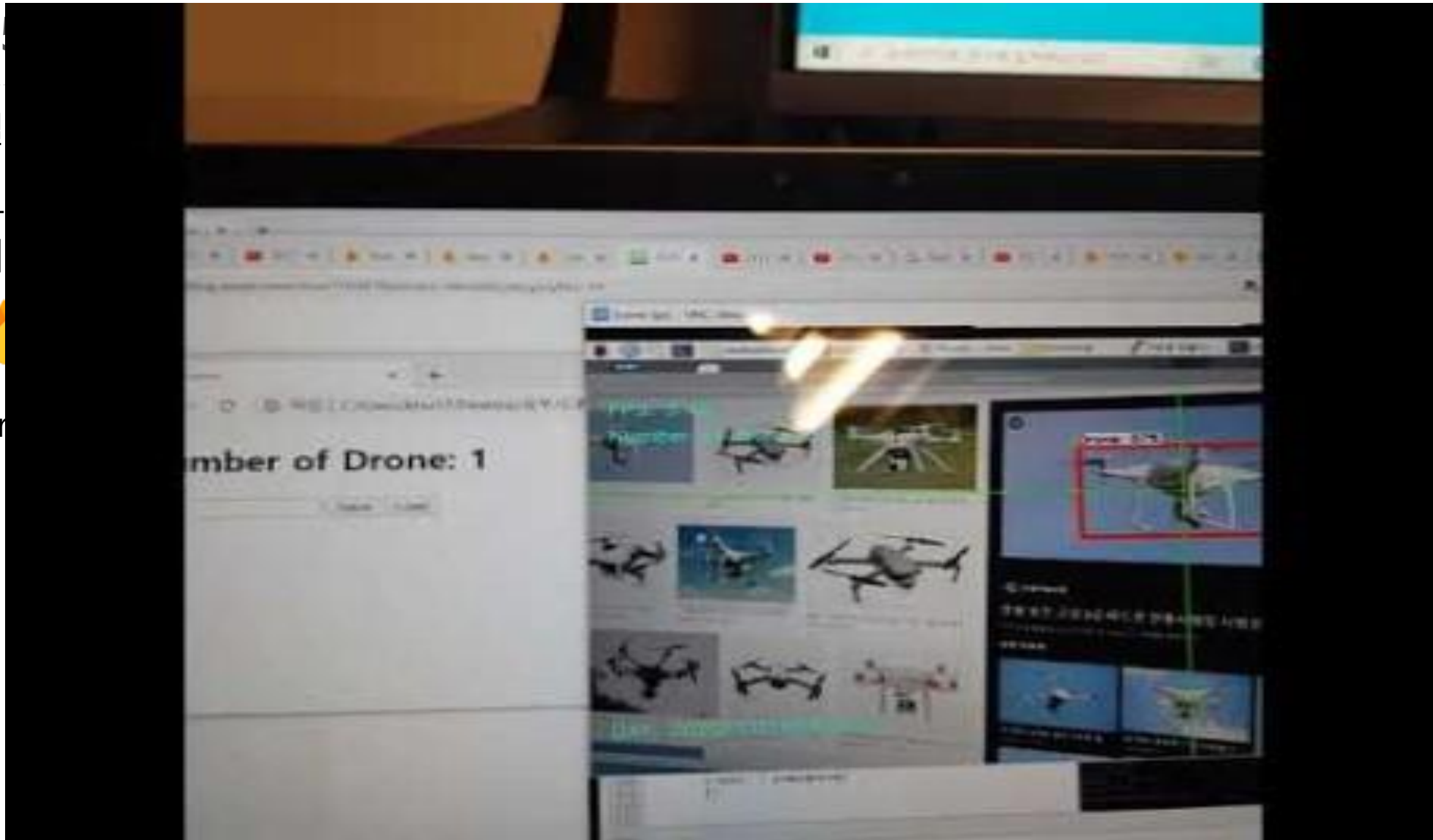
낮은  
라즈베리  
Google  
Tflite



## 05 결과 및 토의

결과 리뷰 – google coral 사용



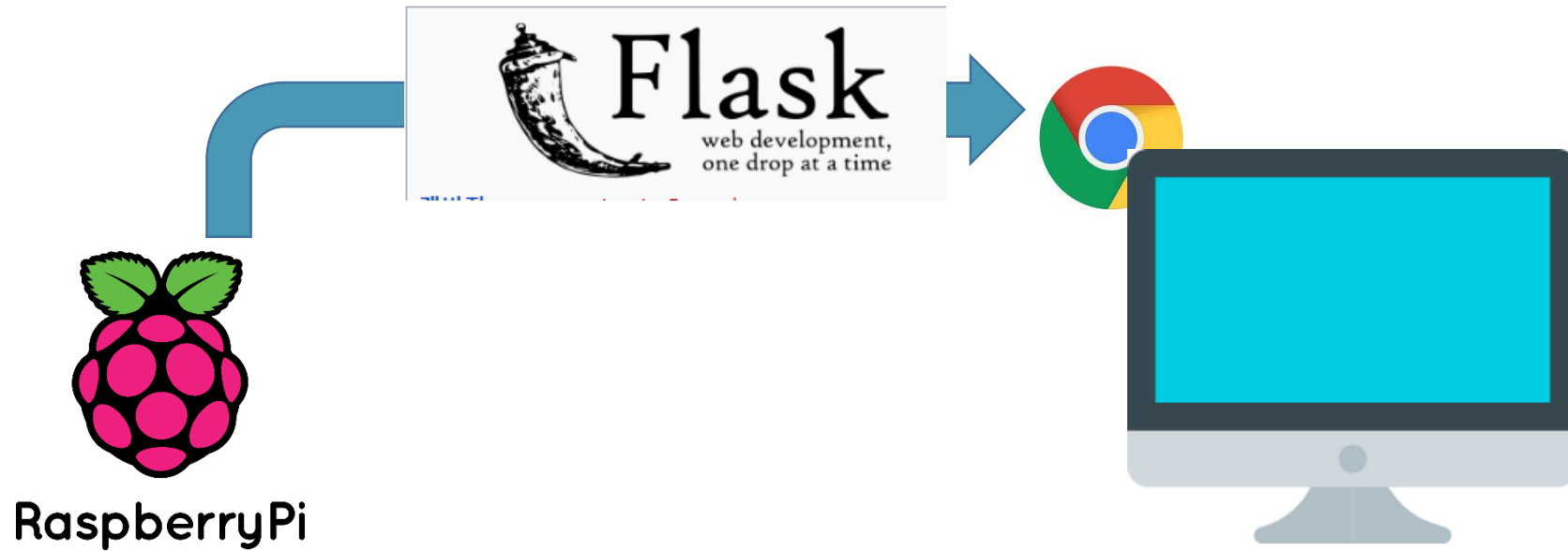


RaspberryPi

## 05 결과 및 토의

### 결과 리뷰 – 실시간 웹캠 스트리밍

라즈베리파이에서 담당 영공에 대한 시야를 사용자에게 실시간 전송 사용자가 영공에 대해 모니터링하고 즉각 반응을 가능하게 한다.



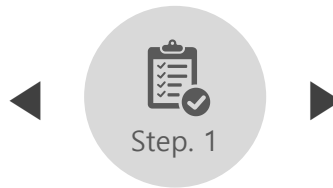
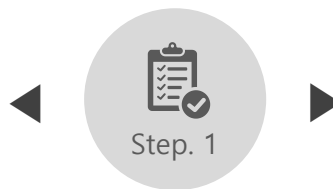
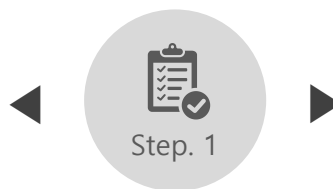
## 05 결과 및 토의

### 문제

YOLOv3를 이용하여 실시간 객체 탐지를 시도하였으나 낮은 컴퓨팅능력의 라즈베리파이에서는 yolo를 통해 실시간 객체 탐지가 불가능 했다.

ssd - mobilenet으로 웹캠 실시간 object detect를 시도하였으나 3~4FPS, 실시간으로 하기위해서는 FPS가 20~30은 나와야한다.

웹에서 실시간 영상 스트리밍



### 해결방안

Tensorflow의 tflite파일은 안드로이드, ios, raspberry pi등에서 사용할수 있도록 하는 비교적 가벼운 파일형태이며 ssd - mobilenet을 통해 구현

Google Coral usb acceleator로 외부 Edge TPU사용하여 Neural Network Model의 Interface 수 행시간을 매우짧게 줄여준다

Python flask 마이크로 웹프레임 워크를 이용하여 구현



## 참고문헌

Yolov3 <https://pjreddie.com/darknet/yolo/>

MobileNet <http://openresearch.ai/t/mobilenets-efficient-convolutional-neural-networks-for-mobile-vision-applications/20>

Flask webcam streaming  
<https://blog.miguelgrinberg.com/post/video-streaming-with-flask>

Create Firestore Database  
[https://www.youtube.com/watch?list=PLI-K7zZEsYLmnJ\\_FpMOZgyg6XclGBu2OX&v=2Vf1D-rUMwE&feature=emb\\_logo](https://www.youtube.com/watch?list=PLI-K7zZEsYLmnJ_FpMOZgyg6XclGBu2OX&v=2Vf1D-rUMwE&feature=emb_logo)

Google Coral usb accelater  
<https://geeksvoyage.com/pi%20news/news-google-coral/>

Opencv –python  
<https://m.blog.naver.com/samsjang/220498694383>