

안티드론

사이버 보안 캡스톤디자인

최종 발표



김태현, 제진명

지도교수 : 최원석 교수님



사이버보안 캡스톤디자인

최종발표

CONTENTS

01

주제 선정 배경

02

프로젝트 목적

03

개발시 요구된 기술 및 SW/HW

04

주요 기능 및 시스템 구성

05

기능 시연

01 주제 선정 배경

사우디 산유량 반토막 낸 드론 테러...美 "이란이 공격했다"

[중앙일보] 입력 2019.09.15 16:34 수정 2019.09.15 16:51

김성락 기자 한애란 기자



예멘 후티 반군의 무인기(드론) 공격으로 불에 휩싸여 있는 사우디아라비아 석유 시설 [로이터=연합뉴스]

사우디아라비아의 주요 석유 시설과 유전이 예멘 반군의 무인기(드론) 공격을 받아 가동을 중단했다. 사우디아라비아의 원유 생산량이 반토막 났다. 유가 급등을 비롯한 국제 원유시장의 혼란이 예상된다.



추천기사



文의 '판문점 동행' 요청
트럼프는 3차례 거절했다

하노이 결렬후 김정은 "먼 길 왔는데"
트럼프 "비행기로 北 바래다주겠다"

엘 클라시코 수만 관중 '드론 폭탄'에 당할 뻔

[중앙일보] 입력 2020.05.20 00:03 수정 2020.05.20 07:20 | 경제 7면 | 지면보기

송지훈 기자



엘 클라시코에서 볼을 다투는 레알 벤텐마(오른쪽)와 바르셀로나 세메도. [로이터=연합뉴스]



추천기사



文의 '판문점 동행' 요청
트럼프는 3차례 거절했다

하노이 결렬후 김정은 "먼 길 왔는데"
트럼프 "비행기로 北 바래다주겠다"

01 주제 선정 배경

글로벌 드론 시장 전망

구분	2020(E)	2025(E)	연평균성장율(CAGR)
국방용 드론	~50억	~75억	~7%
배달용 드론	<10억	~50억	~60%
산업용 드론	~15억	~150억	~50%
총계	~80억	>250억	~30%

[출처:US Equity Research²⁾]

- 드론을 이용한 군사 테러, 주요 시설 테러 등의 사이버 테러가 증가
- 드론 시장 발전 속도에 비해 안티드론 기술 부족
- 사람이 진행하던 예전 방식과 달리 최근 군사 작전에 동원되는 방식은 시스템을 공격하는 방법도 있지만 드론등의 무인 기체를 이용한 공격이 발전되고 있음
- 이와 같은 사이버 공격을 방어하는 시스템이 필요함.

01 주제 선정 배경



국가보안시설의 등급은 중요도 및 파괴·마비·점령시 영향력에 따라 1급, 2급, 3급으로 분류된다.



02 프로젝트 목적

- 드론을 방어하기 위해서는 영공에 나타난 드론을 탐지하는 기술이 첫번째이다.
- 컴퓨터비전과 DEEP LEARNING - CNN을 이용하여 드론비행금지구역의 영공을 로봇이 실시간 모니터링하고 드론이 나타날 시 이를 감지하고 위치와 시간, 사진과 영상을 관제탑에 전송한다면 언제 공격해 올지 모를 드론을 방어하기 위해서 사람이 24시간 직접 관찰할 필요가 없어진다.
- 사람이 경계를 서면 어느 순간 방심할 수 있고 그 틈에 공격이 가능하다.
- 드론 탐지와 실시간 모니터링 알람 서비스와 즉각 반응이 가능한 로봇을 드론비행위험, 금지구역 (원전시설, 군사시설 등과 같은)에 배치하여 그 지역을 지키는 것이다.
- 추가적으로 로봇이 설치된 지역에 드론 출몰 횟수와 시간, 거리 등을 데이터베이스로 축적하고 축적된 빅데이터 분석을 통해 앞으로 일어날 테러를 예측하고 대비할 수 있을 것이라고 기대한다.

03 개발시 요구된 기술 및 SW/HW

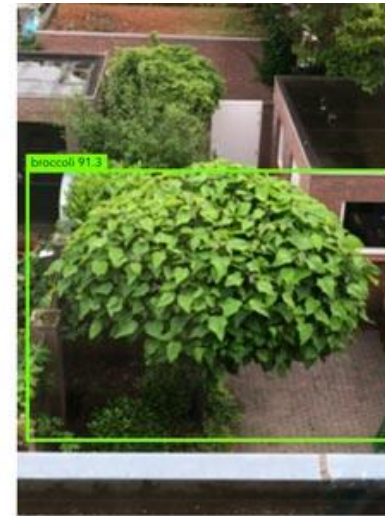
SSD MOBILENET



Correct!



Correct!

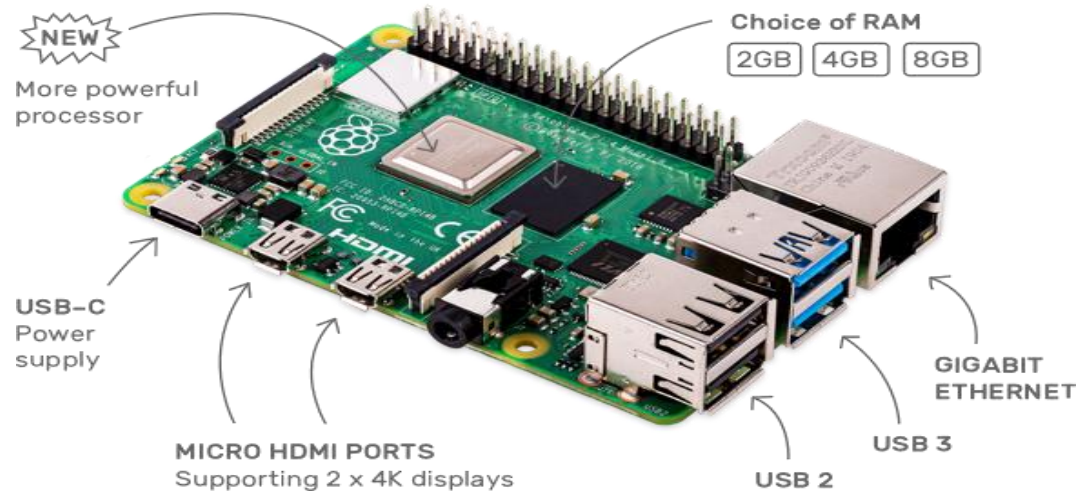


Uuuhh?!

- 하나의 깊은 신경망을 사용하여 이미지에서 물체를 감지하는 방법을 제시
- SSD 모든 계산을 단일 네트워크로 캡슐화하기 때문에 객체 제안이 단순. 이를 통해 SSD를 감지 구성 요소가 필요한 시스템에 쉽게 학습하고 통합
- SSD가 빠른 방법과 정확도를 제공함과 동시에 교육 및 추론을 위한 통합 프레임 워크를 제공. 다른 단일 스테이지 방법과 비교하여 SSD는 입력 이미지 크기가 더 작더라도 정확도가 훨씬 뛰어남.

03 개발시 요구된 기술 및 SW/HW

RASPBERRY PI 4



- Broadcom BCM2711, 쿼드 코어 Cortex-A72 (ARM v8) 64 비트 SoC @ 1.5GHz
- 2GB, 4GB 또는 8GB LPDDR4-3200 SDRAM
- 기가비트 이더넷

03 개발시 요구된 기술 및 SW/HW

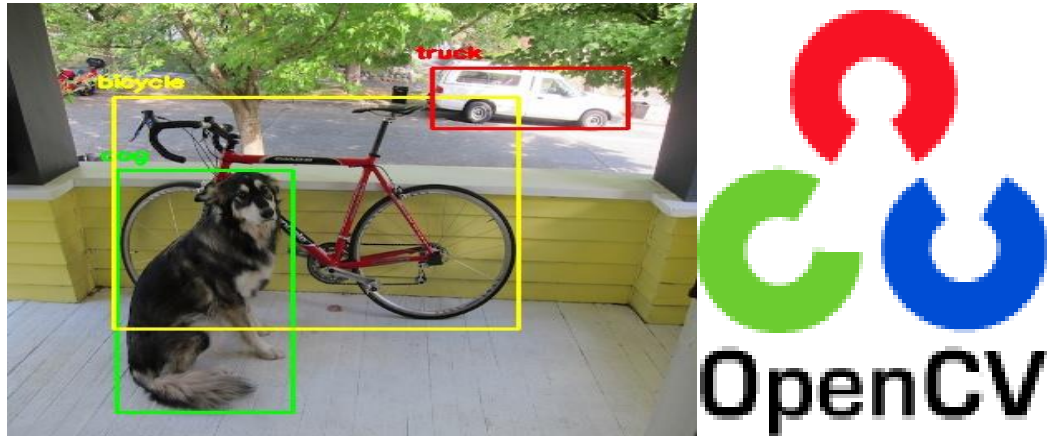
GOOGLE CORAL ACCELATOR



- 구글에서 만든 Machine Learning에 특화된 Edge TPU를 장착한 USB Accelerator.
- 라즈베리파이와 호환 가능.
- Edge TPU를 장착했기 때문에 neural network model의 inference 수행 시간이 매우 짧은 성능을 보여줍니다. TensorFlow lite만 지원
- USB Accelerators는 Edge TPU만 장착한 형태이며, 다른 Single Board Computer와 USB를 연결하여 Machine Learning 연산 가속이 가능

03 개발시 요구된 기술 및 SW/HW

OPEN CV



- 오픈소스 컴퓨터 비전 라이브러리 중 하나로 크로스플랫폼과 실시간 이미지 프로세싱에 중점을 둔 라이브러리
- 딥러닝 관련 연구가 python으로 진행되면서 python 라이브러리의 사용 빈도가 더욱 늘어남
- 영상 관련 라이브러리로서 사실상 표준의 지위를 가지고 있음. 조금이라도 영상처리가 들어간다면 필수적으로 사용하게 되는 라이브러리.

03 개발시 요구된 기술 및 SW/HW

ETC



Firebase



- Firebase : On-line으로 제공되는 유연하고 확장 가능한 NoSQL 클라우드 데이터베이스

- Flask : PYTHON으로 작성된 마이크로 웹 프레임워크



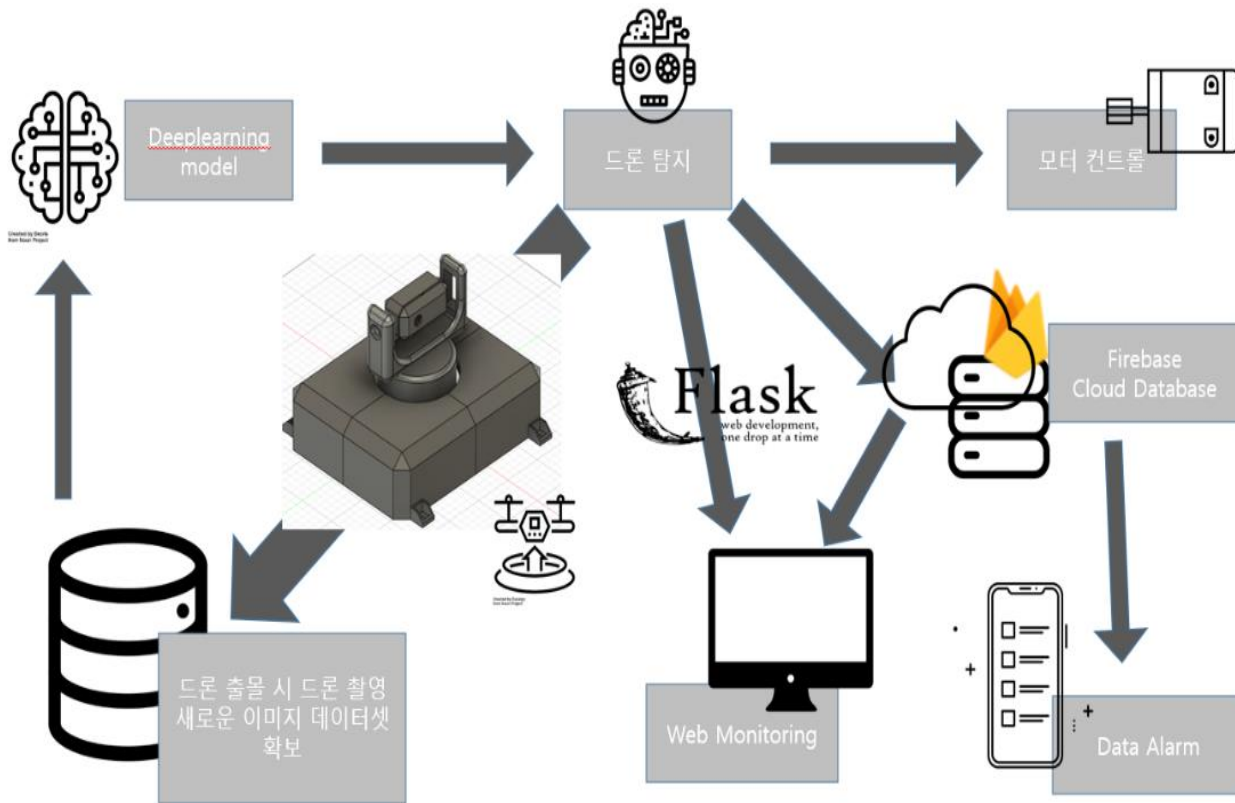
AUTODESK®
FUSION 360™



- Fusion 360 : 오토데스크에서 출시한 제품디자인(product design)에 특화된 3D CAD/CAM/CAE 소프트웨어

- 로지텍 HD웹캠 C270

04 주요 기능 및 시스템 구성



1. 웹캠으로 영상 촬영
2. 딥러닝 모델을 기반으로 드론을 탐지
3. 드론의 기체수, 날짜 정보를 데이터베이스에 업로드
4. 탐지한 드론 객체가 영상의 중심에 위치하도록 2개의 모터를 이용하여 상하 좌우 조절
5. 웹캠으로 촬영한 영상과 데이터베이스에서 가져온 데이터로 웹 스트리밍



04 주요 기능 및 시스템 구성

- CAMERA : 영공의 드론을 탐색하고 영상정보를 RASPBERRY PI로 전송
- RASPBERRY PI : CAMERA에서 가져온 영상 정보를 학습된 딥러닝 모델을 통하여 드론을 식별한다. FLASK 웹 서버를 생성하고 OPEN CV로 가시성을 높인 영상 정보를 스트리밍 할 수 있도록 웹 서비스를 제공한다. 식별된 드론을 쫓아가도록 모터를 제어한다.
- WEB : RASPBERRY PI에서 받은 영상정보를 웹 스트리밍 할 수 있도록 하였다.
- ROBOT : PAN/TILTS 2자유도로 디자인된 로봇은 드론을 효율적으로 탐색할 수 있도록 디자인하였다.

04 주요 기능 및 시스템 구성

```
import cv2
from threading import Thread
import time
import numpy as np

class VideoStream:
    def __init__(self, resolution=(640,480), framerate=30):
        print("init")
        self.stream = cv2.VideoCapture(0)
        (self.grabbed, self.frame) = self.stream.read()
        self.stopped = False
        time.sleep(2.0)

    def start(self):
        print("start thread")
        t = Thread(target=self.update, args=())
        t.daemon = True
        t.start()
        return self
```

```
def update(self):
    print("read")
    while True:
        if self.stopped:
            return

        (self.grabbed, self.frame) = self.stream.read()

    def read(self):
        return self.frame

    def stop(self):
        self.stopped = True
```

RASPBERRY PI에 연결된 웹 캠을 이용하여
영상을 촬영하기 위한 코드

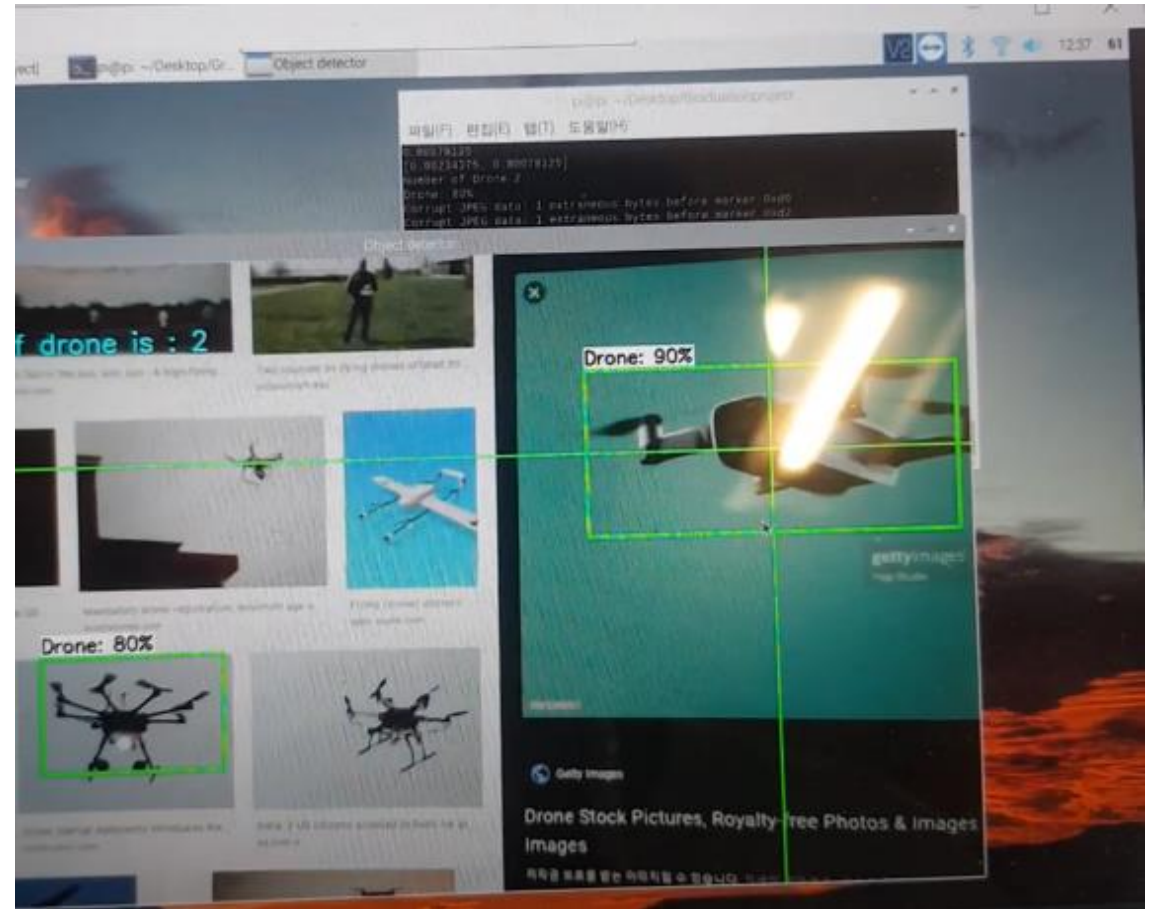
04 주요 기능 및 시스템 구성

```
if len(num) != 0:
    most = num.index(max(num))
    lx = int(max(1, (boxes[most][1] * imW)))
    ly = int(max(1, (boxes[most][0] * imH)))
    lw = int(min(imW, (boxes[most][3] * imW)))
    lh = int(min(imH, (boxes[most][2] * imH)))
    x_medium = int((lx+lw)/2)
    y_medium = int((ly+lh)/2)
    cv2.line(frame, (x_medium, 0), (x_medium, 480), (10, 255, 0), linethickness)
    cv2.line(frame, (0, y_medium), (640, y_medium), (10, 255, 0), linethickness)

for i in range(len(scores)):
    if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

        # Get bounding box coordinates and draw box
        # Interpreter can return coordinates that are outside of image dimensions, need
        ymin = int(max(1, (boxes[i][0] * imH)))
        xmin = int(max(1, (boxes[i][1] * imW)))
        ymax = int(min(imH, (boxes[i][2] * imH)))
        xmax = int(min(imW, (boxes[i][3] * imW)))

        cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), rectangle_color, boxthickness)
```



04 주요 기능 및 시스템 구성

D = 실제거리 x = 바운딩박스변의길이

m = 기울기

$D = mx + c$

$D(1) = 50\text{cm}$

$x(1) = x_{\max}(1) - x_{\min}(1) = 160$

$D(2) = 120\text{cm}$

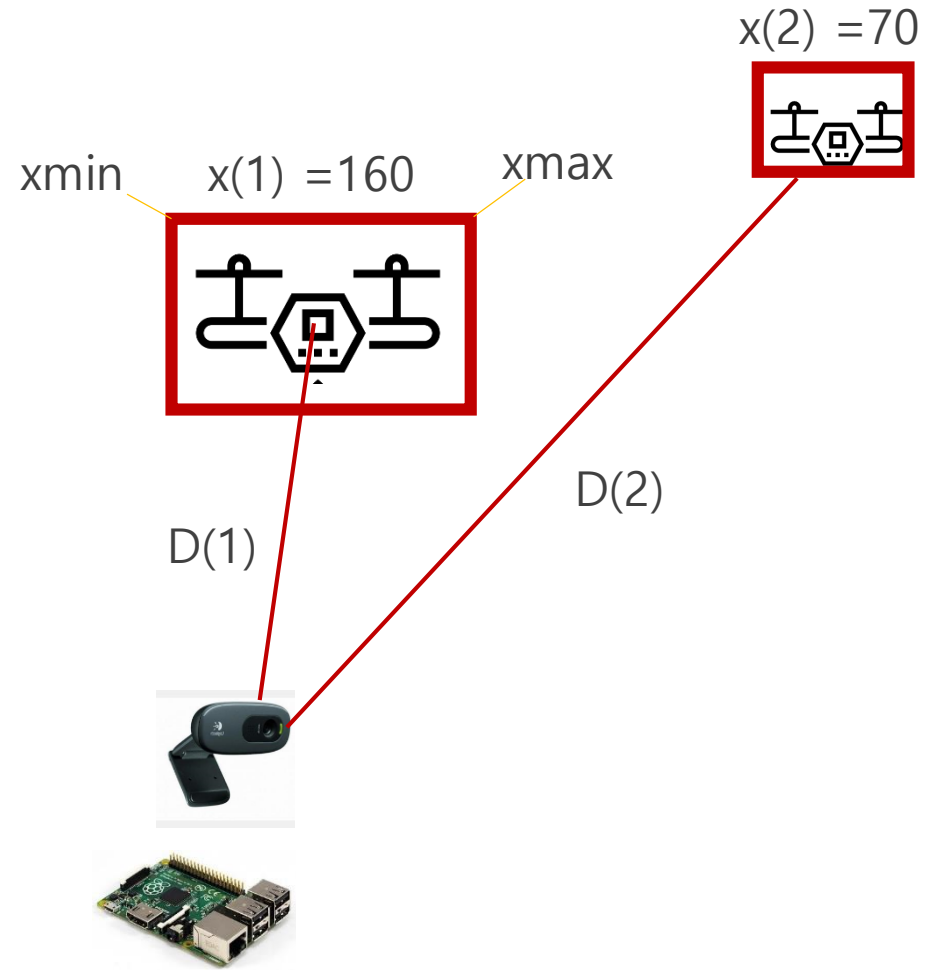
$x(2) = x_{\max}(2) - x_{\min}(2) = 70$

$50 = 160 + C(1)$

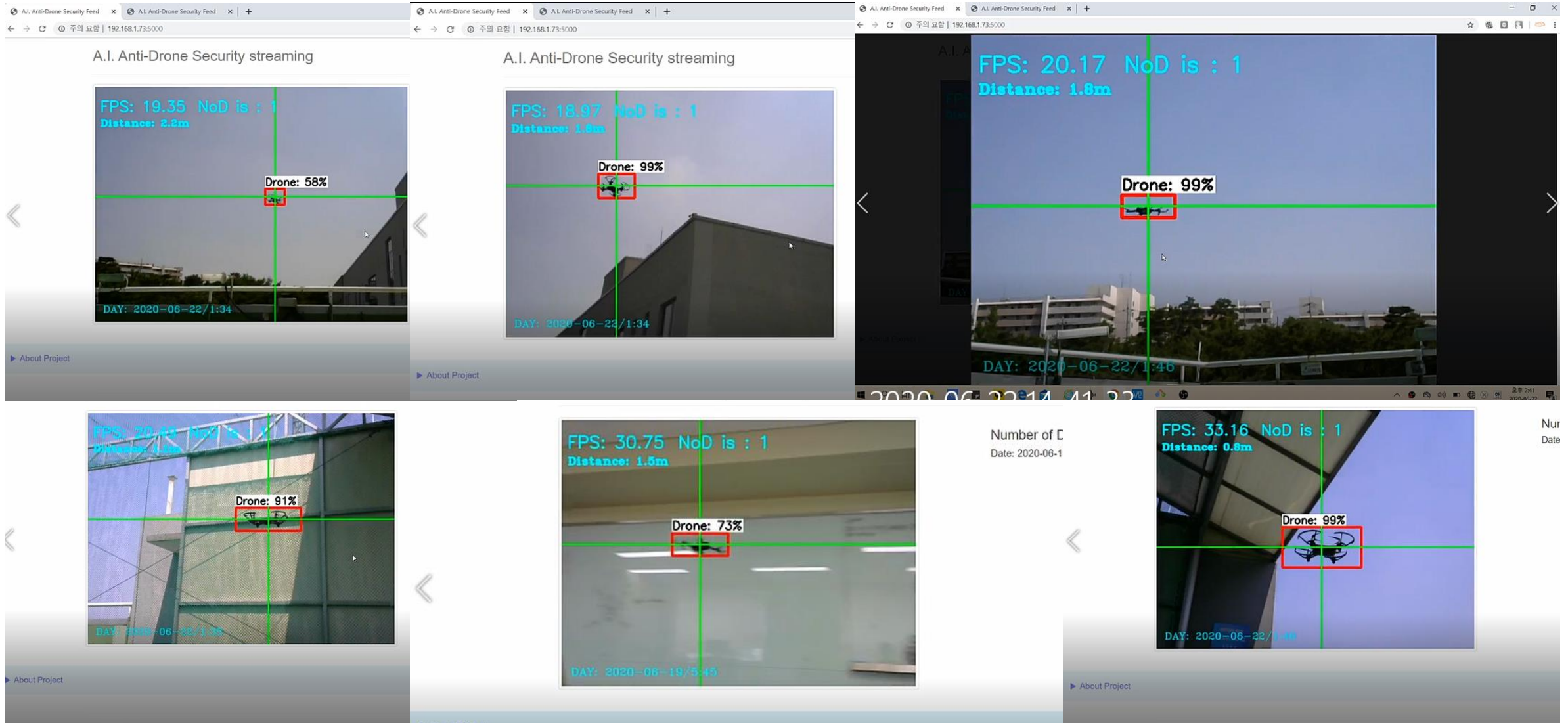
$120 = 70 + C(2)$

$M = -7/9 \quad C = 1570/9$

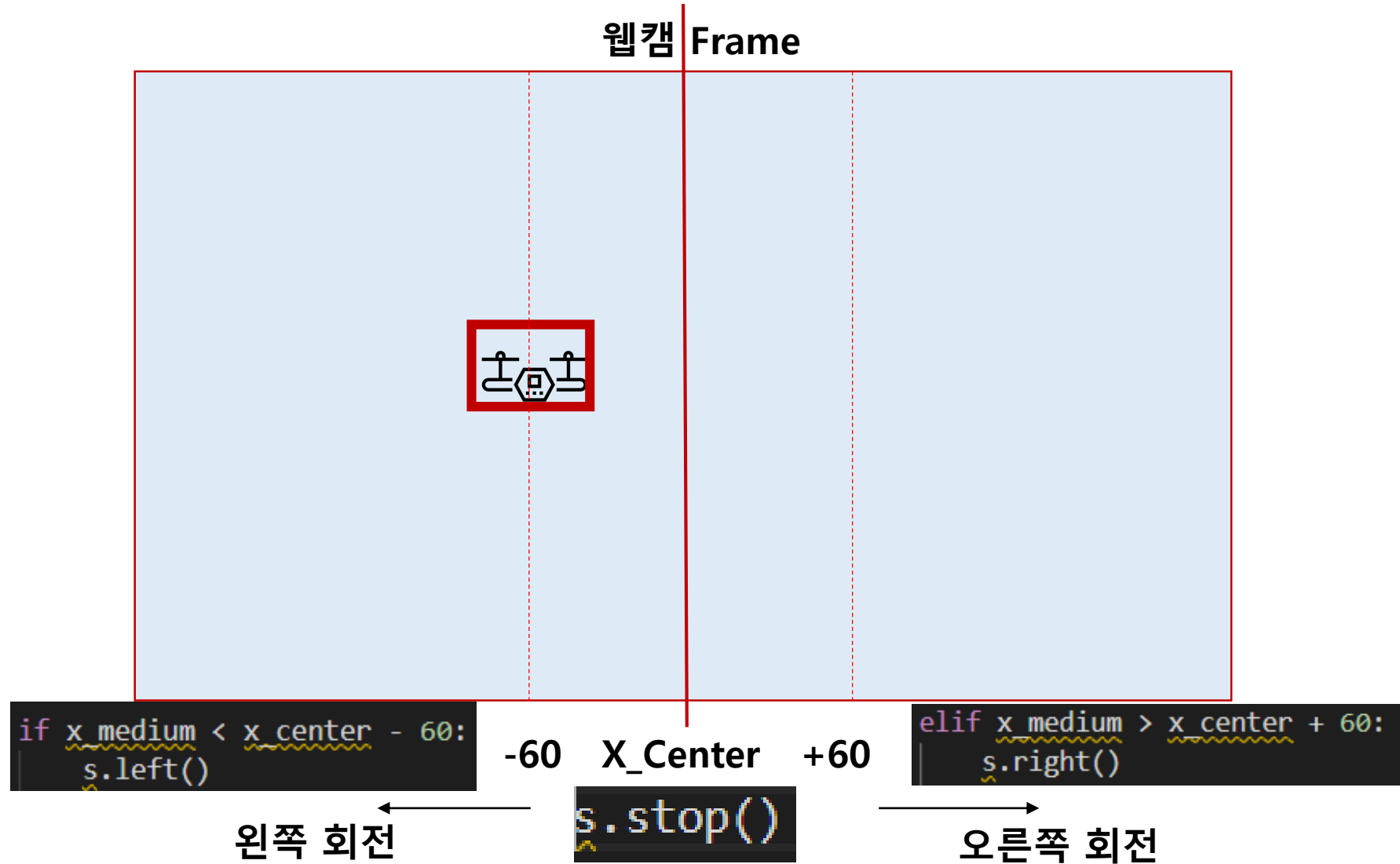
$Y = -(7/9)x + 1570/9$



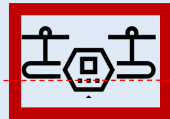
04 주요 기능 및 시스템 구성



04 주요 기능 및 시스템 구성



04 주요 기능 및 시스템 구성



```
elif y_medium > y_center + 30:  
    s.down()
```

+30

위로

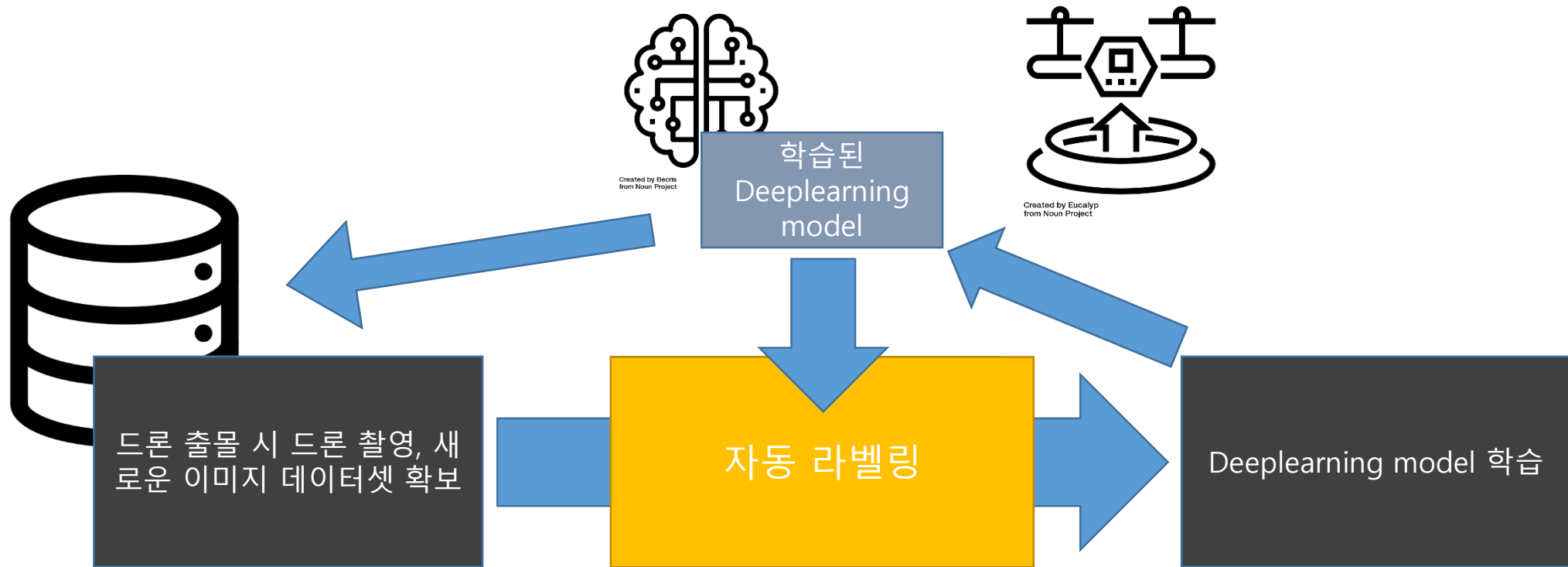
Y_Center

-30

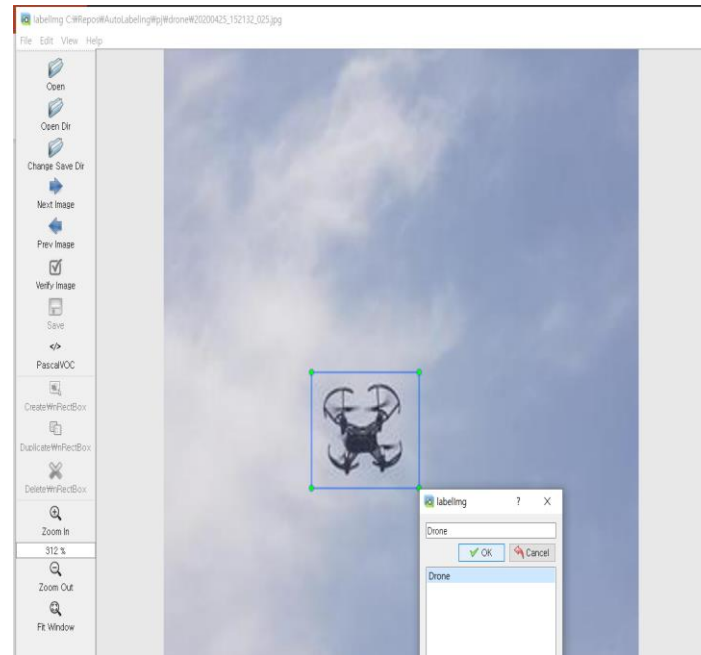
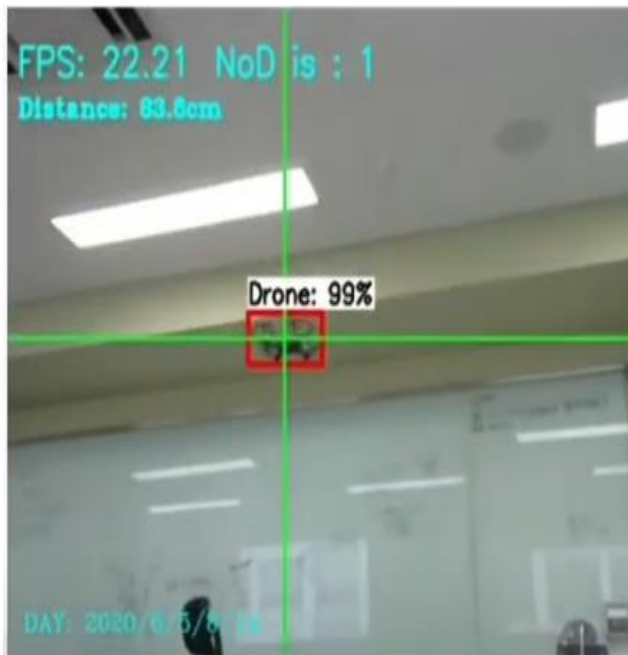
아래로

```
if y_medium < y_center - 30:  
    s.up()
```

04 주요 기능 및 시스템 구성



04 주요 기능 및 시스템 구성



✓ 0 - 9 (3)



20200425_15213
2_025



20200425_15213
2_025

04 주요 기능 및 시스템 구성

```
allimagecount = failcount + successcount
if len(num) == 0:
    failcount += 1
    print('fail: {} processing: {}'.format(failcount, allimagecount))
    time.sleep(0.1)
    shutil.move(PATH_TO_IMAGES+'\\'+filename, failimages+'\\'+filename)

if len(num) != 0:
    successcount += 1
    print('success: {} processing: {}'.format(successcount, allimagecount))
    shutil.move(PATH_TO_IMAGES+'\\'+filename, successimages+'\\'+filename)
# All the results have been drawn on the image, now display the image
#cv2.imshow('filename', image)

end_time = time.time()
process_time = end_time - start_time
processingtime = processingtime + process_time
print("===== A frame took {:.3f} seconds=====Whole time {:.3f} seconds({:.1f}min)=====".format(process_time))
# Press any key to continue to next image, or press 'q' to quit
if cv2.waitKey(0) == ord('q'):
    break

# Clean up
cv2.destroyAllWindows()

print('End-----')
print('Fail Number of Image : ', failcount)
print('Success Number of Image : ', successcount)
print('All Number of Image : ', allimagecount)
```

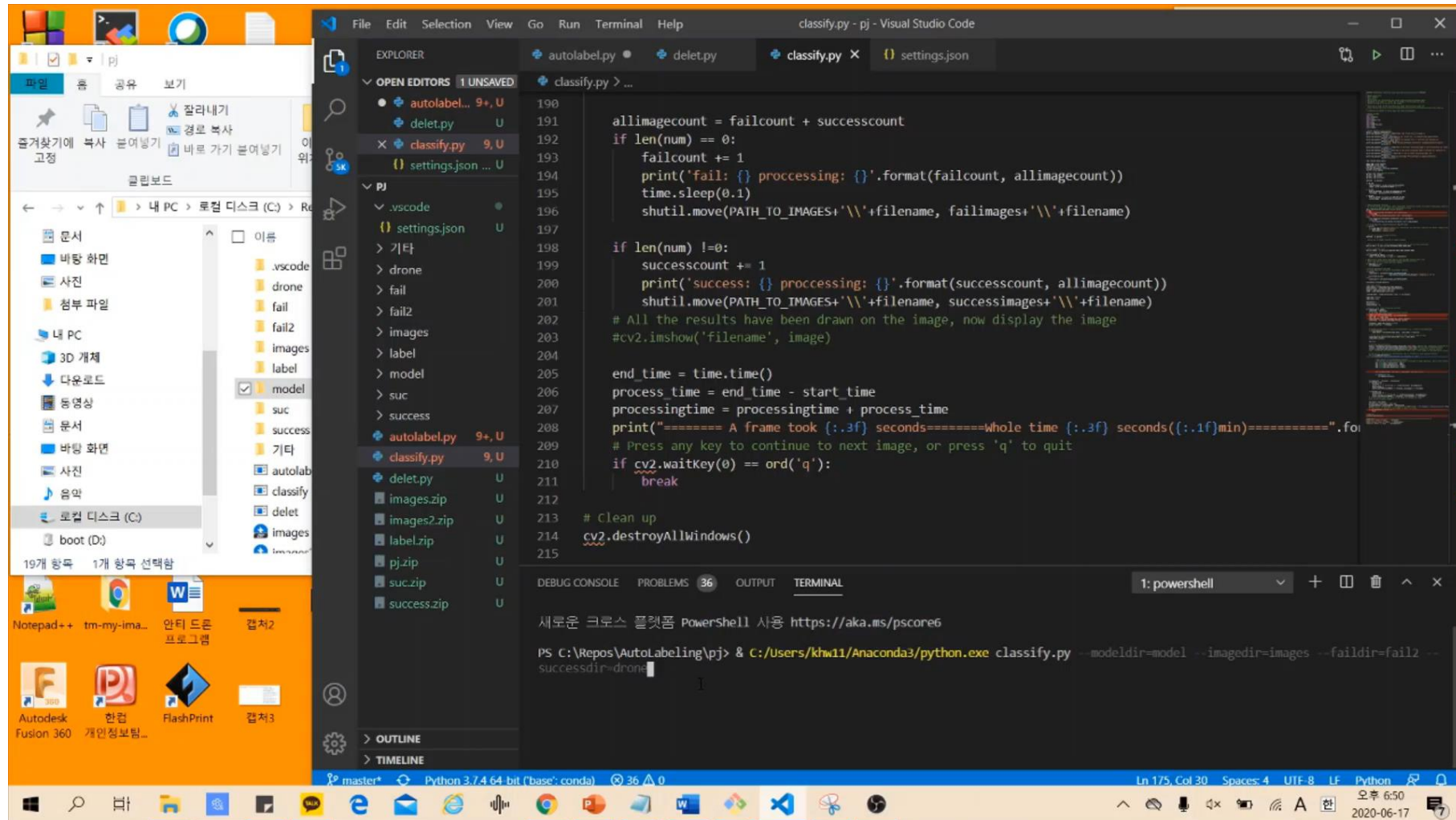
```
end_time = time.time()
process_time = end_time - start_time
print("===== A frame took {:.3f} seconds=====".format(process_time))
print('Choose Success Or Fail')

if cv2.waitKey(0) == ord('d'):
    f = open(successfolder+'\\'+filename, 'w')
    f.write('fail')
    f.close
    print('you pressed d')
    shutil.move(successfolder+'\\'+filename, failimages+'\\'+filename)
    time.sleep(0.1)
    print("-----sendtofailfoder-----")
    os.remove(successfolder+'\\'+filename3)
    print('delete')
    time.sleep(0.1)
    print("fail")
    cv2.destroyAllWindows()

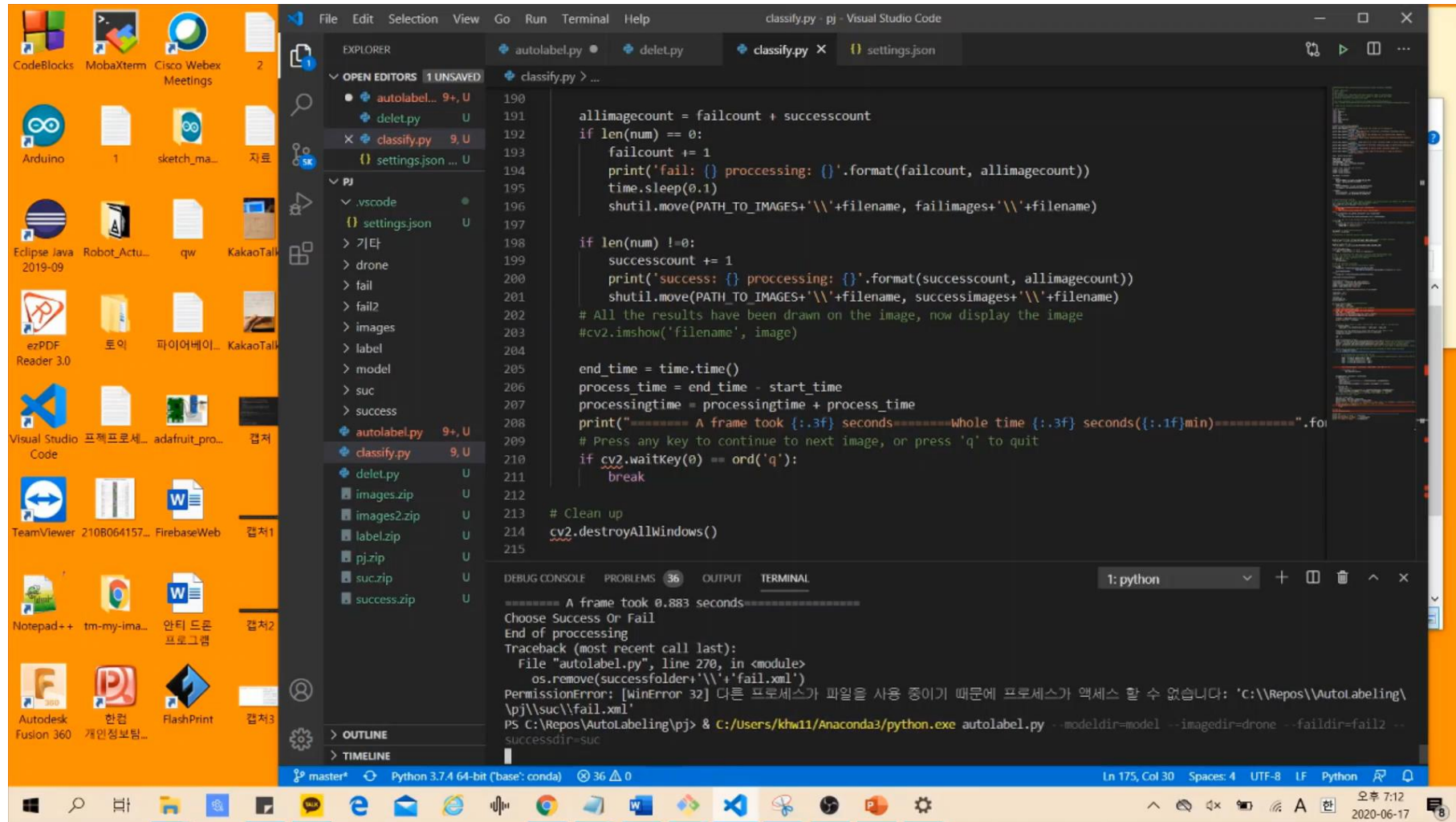
# Press any key to continue to next image, or press 'q' to quit
if cv2.waitKey(0) == ord('q'):
    f = open(successfolder+'\\'+filename, 'w')
    f.write('fail')
    f.close
    print('End of processing ')
    shutil.move(successfolder+'\\'+filename, PATH_TO_IMAGES+'\\'+filename)
    time.sleep(0.1)
    os.remove(successfolder+'\\'+filename3)
    time.sleep(0.1)
    break

# Clean up
cv2.destroyAllWindows()
os.remove(successfolder+'\\'+filename)
print('all end')
```

04 주요 기능 및 시스템 구성



04 주요 기능 및 시스템 구성





05 기능 시연