

# OpenCV와 IoT 기술을 이용한 길고양이 퇴치 시스템

정보통신공학과

20181645 신동운

20181675 원석찬

20181688 정재인

지도교수: 이충호

- 
1. 연구 배경 및 목적
  2. 계획대비 진행상황
  3. 역할분담
  4. 향후 계획
  5. 참고 문헌

# 연구 배경 및 목적

---

## 연구 배경



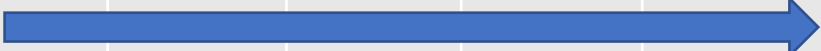
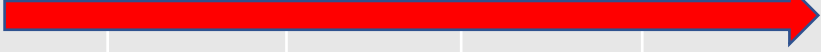
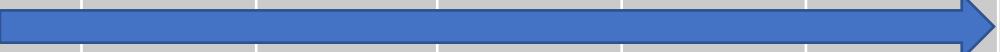

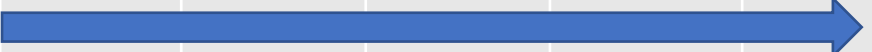

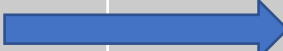

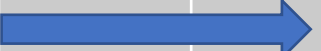

- 길고양이를 퇴치하는 '전기 펜스 설치', '담장 위 장애물 설치', '동작 감지 스프링 쿨러' 등의 방법들은 복잡하고 고비용을 요구함
- 학교 내에 길고양이 퇴치 시스템이 충분하지 않음

## 연구 목적

- 길고양이의 건물 내 침입방지를 위한 간단한 시스템을 OpenCV와 아두이노를 이용하여 구현

# 계획대비 진행상황

4

연구내용	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
연구 계획 수립 및 프로그램 학습	 									
영상처리 코드 제작		 								
아두이노, 라즈베리 파이를 이용한 IOT 제작			 							
실제 시스템 실험 및 수정 보완					 					
보고서 작성 및 발표		 			 					

# 구성원의 역할 분담

5

## 신동운

- 영상처리 코드 개발
- 인공지능 시스템 개발

## 원석찬

- 아두이누 모듈 연결
- 아두이노 코드 개발

## 정재인

- 아두이노 모듈 연결
- 영상처리 연구

공통: 보고서 작성, 발표준비

# 개발 진행상황

6

```
import cv2
import cvlib as cv
from cvlib.object_detection import draw_bbox
import pafy

url = 'https://www.youtube.com/watch?v=E8V02ArInjY&t=2s'

video = pafy.new(url)
best_stream = video.getbest()
cap = cv2.VideoCapture()
cap.open(best_stream.url)

while True:
    ret, frame = cap.read()

    bbox, label, conf = cv.detect_common_objects(frame)

    selected_bbox = []
    selected_label = []
    selected_conf = []

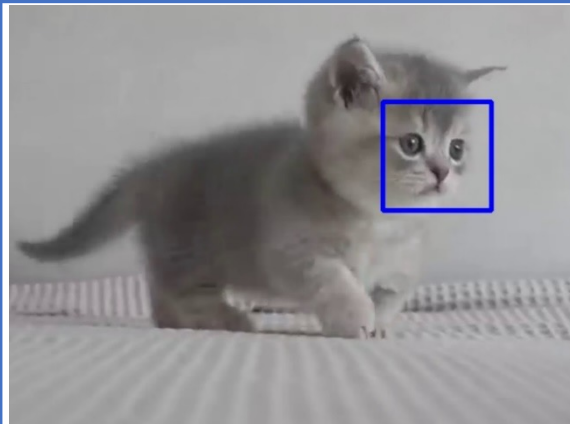
    for idx, obj_label in enumerate(label):
        if obj_label == 'cat':
            selected_bbox.append(bbox[idx])
            selected_label.append(obj_label)
            selected_conf.append(conf[idx])

    output_image = draw_bbox(frame, selected_bbox, selected_label, selected_conf)

    cv2.imshow('Cat Detection', output_image)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

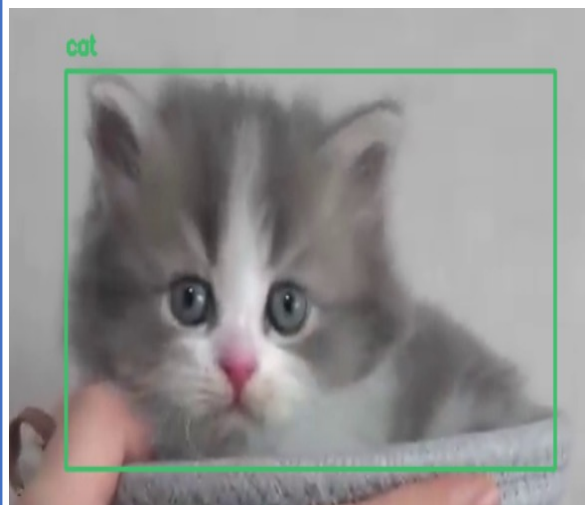
cap.release()
cv2.destroyAllWindows()
```



얼굴 인식이 정확하지 않음



cvlib 라이브러리 추가, haarcascade 제거



- 물체의 이름 출력
- 좀 더 정확하게 인식

# 개발 진행상황

7

```
import cv2
import cvlib as cv
from cvlib.object_detection import draw_bbox
import pafy

url = 'https://www.youtube.com/watch?v=VsBd19tFhFk'
video = pafy.new(url)
best_stream = video.getbest()
cap = cv2.VideoCapture()
cap.open(best_stream.url)

target_classes = ['cat', 'dog', 'bird', 'horse', 'elephant', 'sheep', 'cow', 'giraffe', 'zebra', 'bear']

while True:
    ret, frame = cap.read()

    bbox, label, conf = cv.detect_common_objects(frame)

    selected_bbox = []
    selected_label = []
    selected_conf = []

    for idx, obj_label in enumerate(label):
        if obj_label in target_classes:
            selected_bbox.append(bbox[idx])
            selected_label.append(obj_label)
            selected_conf.append(conf[idx])

    output_image = draw_bbox(frame, selected_bbox, selected_label, selected_conf)

    cv2.imshow('Object Detection', output_image)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```



target\_classes를 만들어  
다양한 동물도 관측 가능

# 개발 진행상황

8

Wifi\_ESP32cam.ino

```
1  #include <WebServer.h>
2  #include <WiFi.h>
3  #include <esp32cam.h>
4
5  const char* WIFI_SSID = "HANBAT_LIB";
6  const char* WIFI_PASS = "";
7
8  WebServer server(80); //servidor en el puerto 80
9
10 static auto loRes = esp32cam::Resolution::find(320, 240); //baja resolucio
11 static auto hiRes = esp32cam::Resolution::find(800, 600); //alta resolucio
12
13 void
14 serveJpg() //captura imagen .jpg
15 {
16     auto frame = esp32cam::capture();
17     if (frame == nullptr) {
18         Serial.println("CAPTURE FAIL");
19         server.send(503, "", "");
20         return;
21     }
22     Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
23         static_cast<int>(frame->size()));
24
25     server.setContentLength(frame->size());
26     server.send(200, "image/jpeg");
27     WiFiClient client = server.client();
28     frame->writeTo(client); // y envia a un cliente (en este caso sera python)
29 }
30
31 void
32 handleJpgLo() //permite enviar la resolucio de imagen baja
33 {
34     if (!esp32cam::Camera.changeResolution(loRes)) {
35         Serial.println("SET-LO-RES FAIL");
36     }
37     serveJpg();
38 }
39
40 void
41 handleJpgHi() //permite enviar la resolucio de imagen alta
42 {
43     if (!esp32cam::Camera.changeResolution(hiRes)) {
44         Serial.println("SET-HI-RES FAIL");
45     }
46     serveJpg();
47 }
48
49 void
50 setup()
51 {
52     Serial.begin(115200);
53     Serial.println();
54
55 }
```

Wifi\_ESP32cam.ino

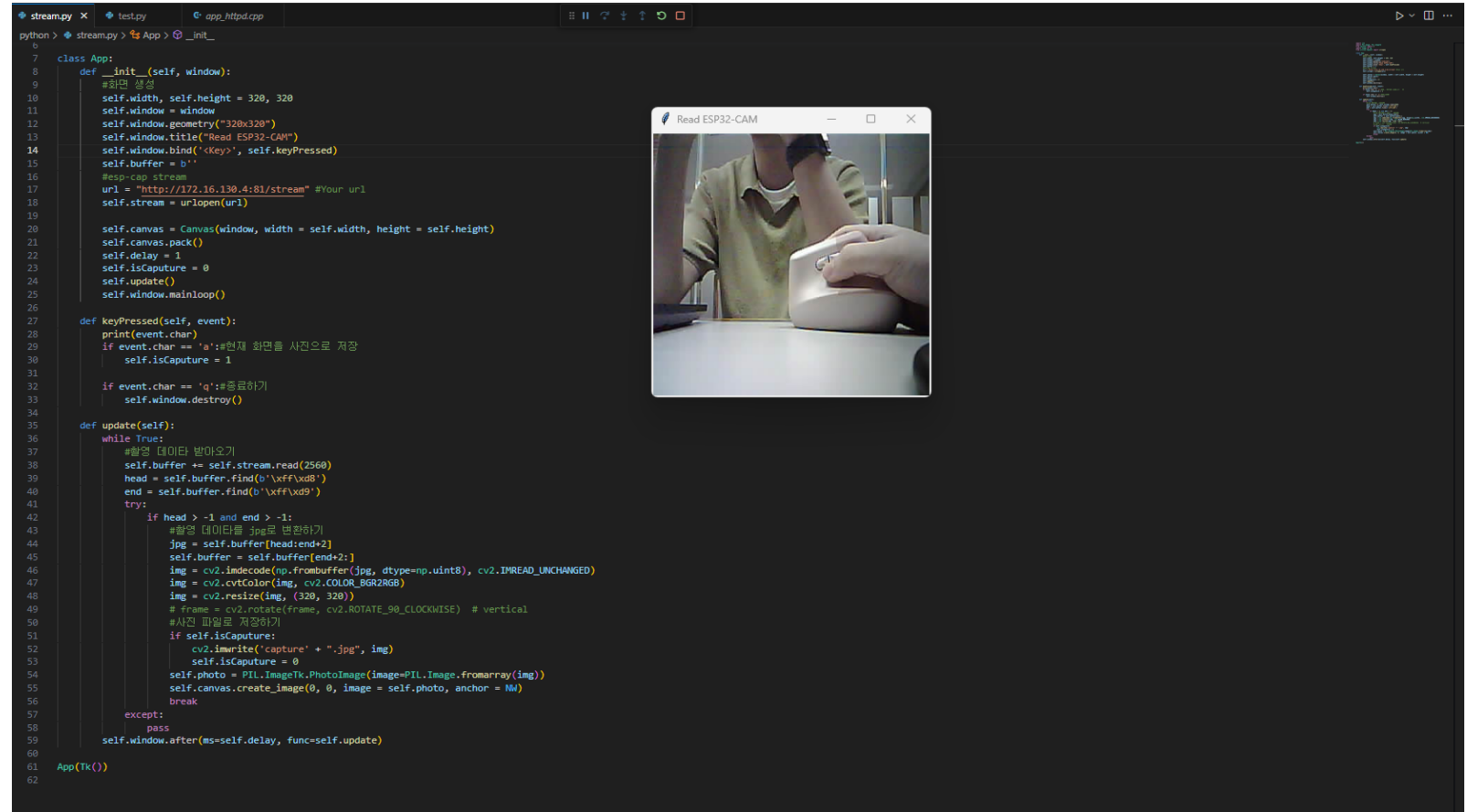
```
38 }
39
40 void
41 handleJpgHi() //permite enviar la resolucio de imagen alta
42 {
43     if (!esp32cam::Camera.changeResolution(hiRes)) {
44         Serial.println("SET-HI-RES FAIL");
45     }
46     serveJpg();
47 }
48
49 void
50 setup()
51 {
52     Serial.begin(115200);
53     Serial.println();
54
55     {
56         using namespace esp32cam;
57         Config cfg;
58         cfg.setPins(pins::AiThinker);
59         cfg.setResolution(hiRes);
60         cfg.setBufferCount(2);
61         cfg.setJpeg(80);
62
63         bool ok = Camera.begin(cfg);
64         Serial.println(ok ? "CAMARA OK" : "CAMARA FAIL");
65     }
66
67     WiFi.persistent(false);
68     WiFi.mode(WIFI_STA);
69     WiFi.begin(WIFI_SSID, WIFI_PASS); //nos conectamos a la red wifi
70     while (WiFi.status() != WL_CONNECTED) {
71         delay(500);
72     }
73
74     Serial.print("http://");
75     Serial.print(WiFi.localIP());
76     Serial.println("/cam-lo.jpg");//para conectarnos IP res baja
77
78     Serial.print("http://");
79     Serial.print(WiFi.localIP());
80     Serial.println("/cam-hi.jpg");//para conectarnos IP res alta
81
82     server.on("/cam-lo.jpg",handleJpgLo);//enviamos al servidor
83     server.on("/cam-hi.jpg", handleJpgHi);
84
85     server.begin();
86 }
87
88 void loop()
89 {
90     server.handleClient();
91 }
92
```

Wifi를 통해 카메라 모듈 연결



# 개발 진행상황

9

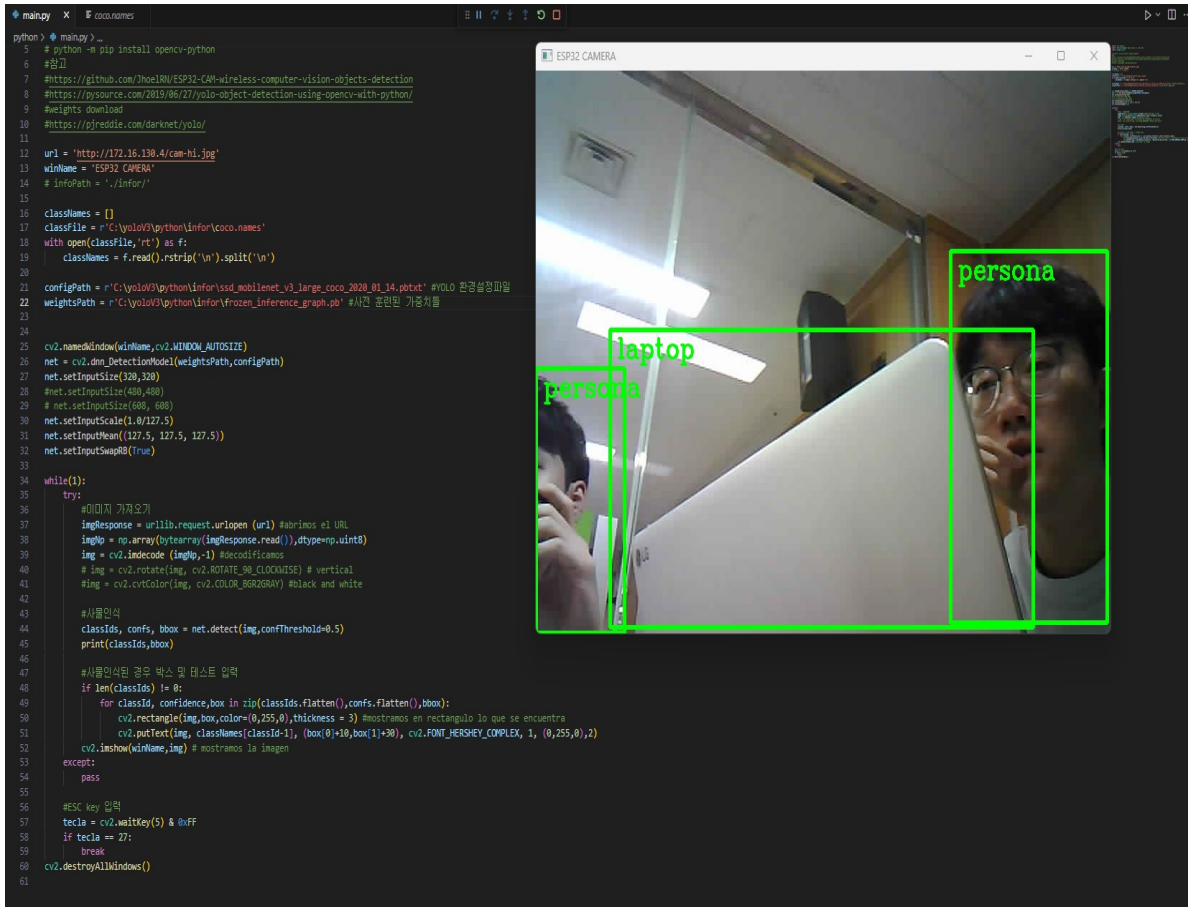


ESP32 CAM을 단독 개체로 분리

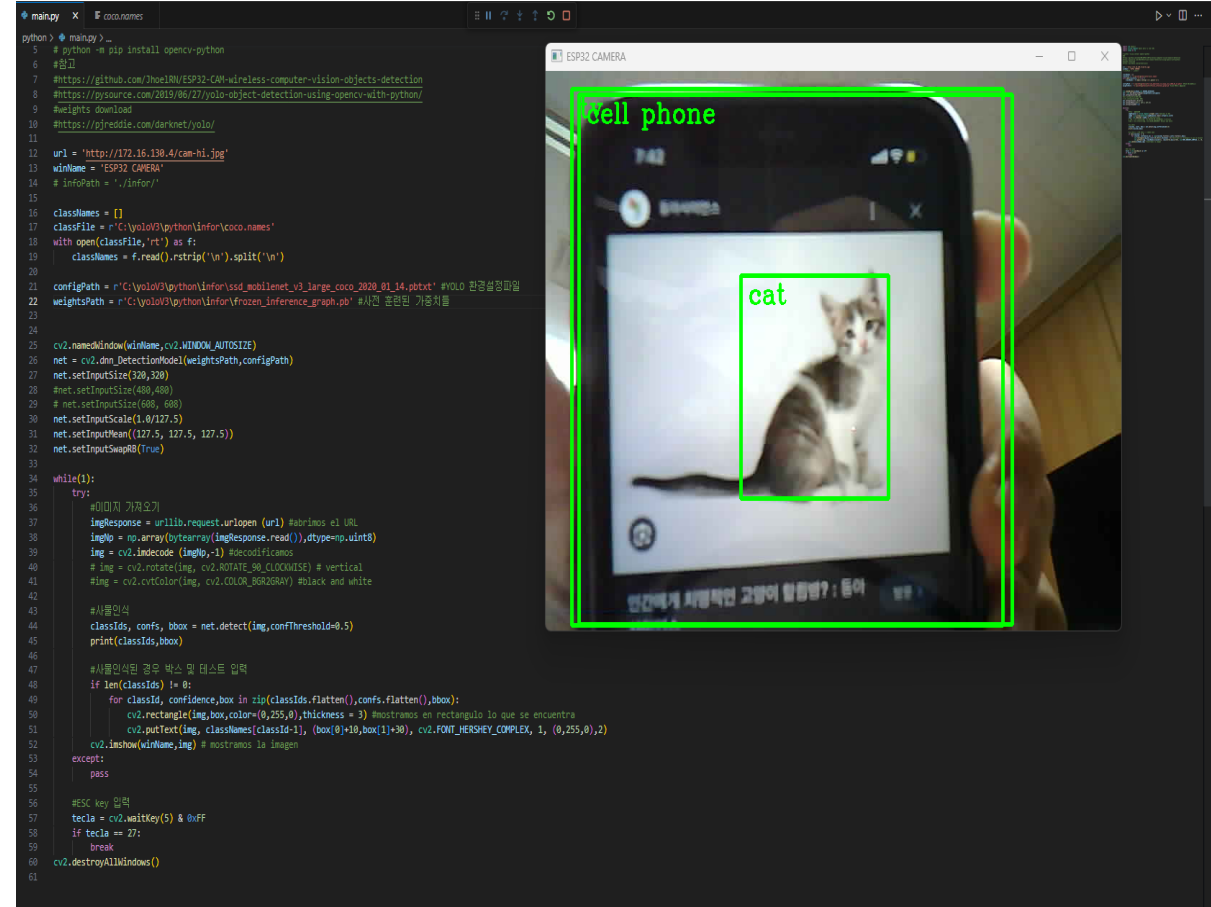
- 아두이노와 카메라 모듈 연결 후 정상 동작 확인
- 파이썬 코드와 아두이노 코드 연결

# 개발 진행상황

10



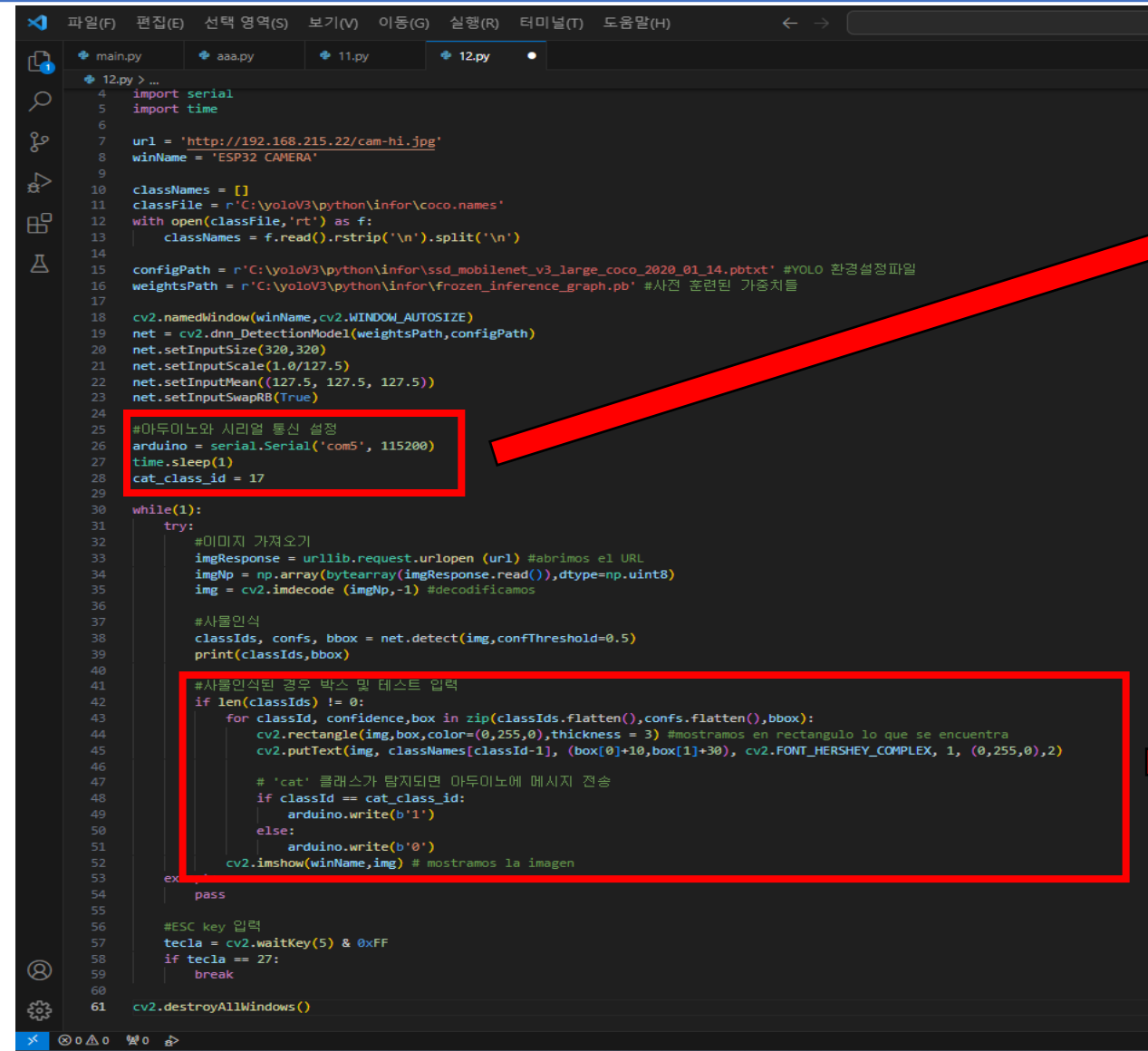
Yolo를 이용하여 객체인식 확인



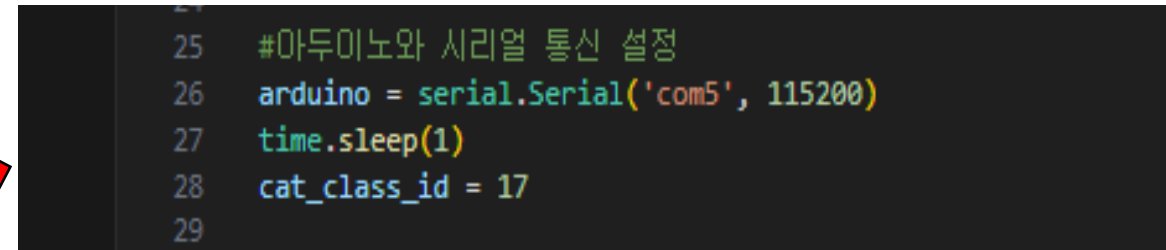
Yolo를 이용하여 고양이 인식 확인

# 개발 진행상황

11



```
4 import serial
5 import time
6
7 url = 'http://192.168.215.22/cam-hi.jpg'
8 winName = 'ESP32 CAMERA'
9
10 classNames = []
11 classFile = r'C:\yoloV3\python\infor\coco.names'
12 with open(classFile, 'rt') as f:
13     classNames = f.read().rstrip('\n').split('\n')
14
15 configPath = r'C:\yoloV3\python\infor\ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt' #YOLO 환경설정파일
16 weightsPath = r'C:\yoloV3\python\infor\frozen_inference_graph.pb' #사전 훈련된 가중치들
17
18 cv2.namedWindow(winName, cv2.WINDOW_AUTOSIZE)
19 net = cv2.dnn_DetectionModel(weightsPath, configPath)
20 net.setInputSize(320, 320)
21 net.setInputScale(1.0/127.5)
22 net.setInputMean((127.5, 127.5, 127.5))
23 net.setInputSwapRB(True)
24
25 #아두이노와 시리얼 통신 설정
26 arduino = serial.Serial('com5', 115200)
27 time.sleep(1)
28 cat_class_id = 17
29
30 while(1):
31     try:
32         #이미지 가져오기
33         imgResponse = urllib.request.urlopen(url) #abrimos el URL
34         imgNp = np.array(bytearray(imgResponse.read()), dtype=np.uint8)
35         img = cv2.imdecode(imgNp, -1) #decodificamos
36
37         #사물인식
38         classIds, confs, bbox = net.detect(img, confThreshold=0.5)
39         print(classIds, bbox)
40
41         #사물인식된 경우 박스 및 테스트 입력
42         if len(classIds) != 0:
43             for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
44                 cv2.rectangle(img, box, color=(0, 255, 0), thickness=3) #mostramos en rectangulo lo que se encuentra
45                 cv2.putText(img, classNames[classId-1], (box[0]+10, box[1]+30), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
46
47                 # 'cat' 클래스가 탐지되면 아두이노에 메시지 전송
48                 if classId == cat_class_id:
49                     arduino.write(b'1')
50                 else:
51                     arduino.write(b'0')
52                 cv2.imshow(winName, img) # mostramos la imagen
53
54         except:
55             pass
56
57         #ESC key 입력
58         tecla = cv2.waitKey(5) & 0xFF
59         if tecla == 27:
60             break
61     cv2.destroyAllWindows()
```



```
25 #아두이노와 시리얼 통신 설정
26 arduino = serial.Serial('com5', 115200)
27 time.sleep(1)
28 cat_class_id = 17
29
```

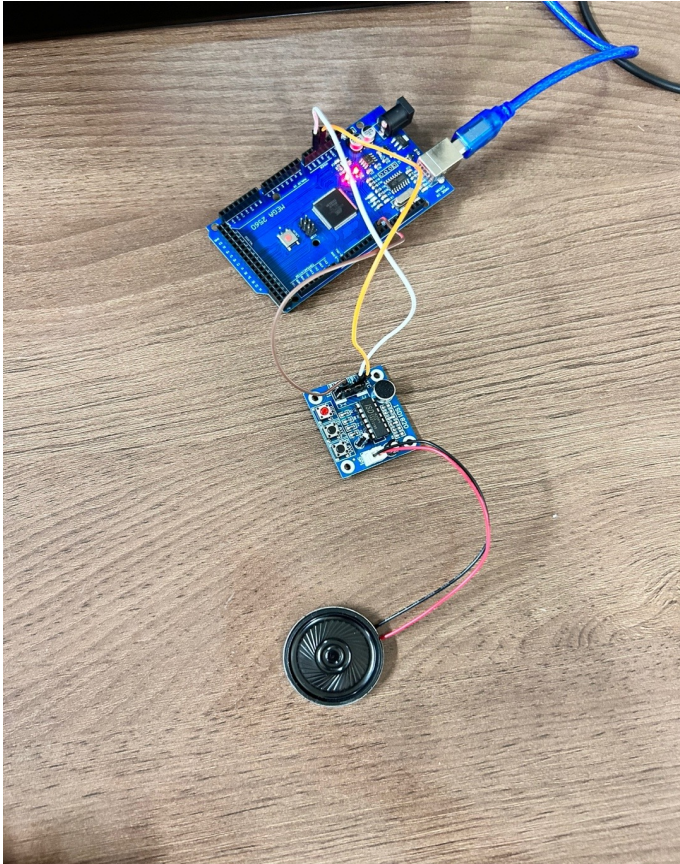
OpenCV와 아두이노를 연결하는 시리얼 통신 설정



```
41 #사물인식된 경우 박스 및 테스트 입력
42 if len(classIds) != 0:
43     for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
44         cv2.rectangle(img, box, color=(0, 255, 0), thickness=3) #mostramos en rectangulo lo que se encuentra
45         cv2.putText(img, classNames[classId-1], (box[0]+10, box[1]+30), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
46
47         # 'cat' 클래스가 탐지되면 아두이노에 메시지 전송
48         if classId == cat_class_id:
49             arduino.write(b'1')
50         else:
51             arduino.write(b'0')
52         cv2.imshow(winName, img) # mostramos la imagen
53
54     except:
55         pass
```

고양이가 감지 되었을 때 아두이노에 신호 전송





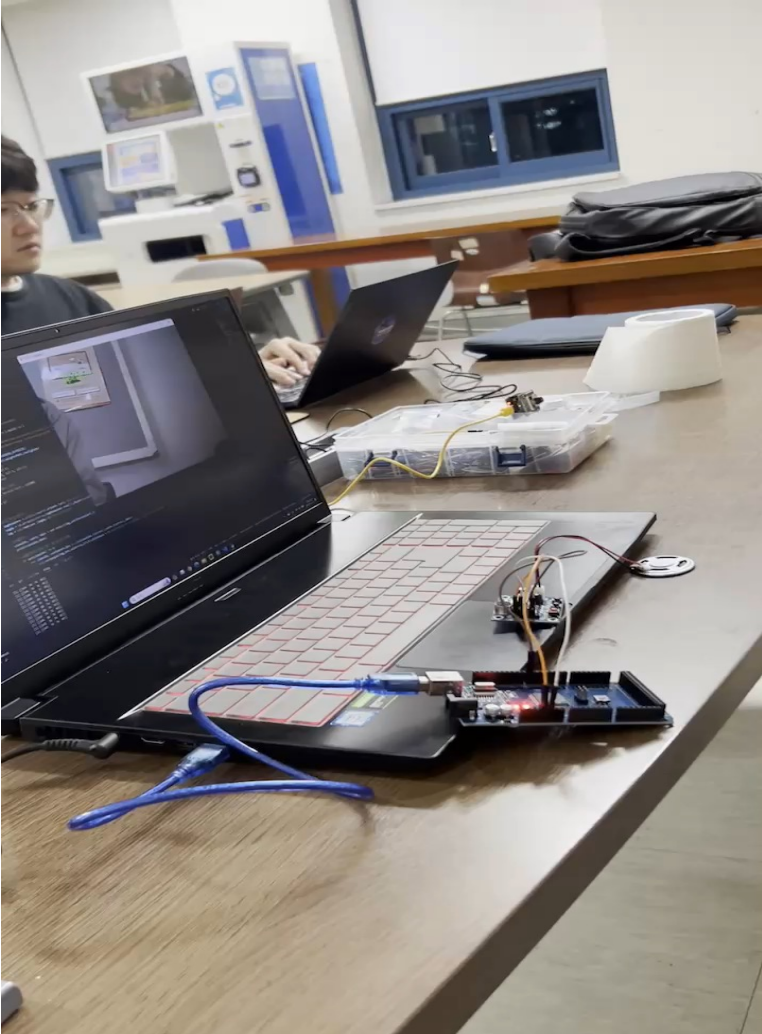
소리 모듈과 아두이노 연결

sketch\_oct13a.ino

```
1  int playe = 7;
2  int data;
3
4  void setup()
5  {
6    Serial.begin(115200);
7    //pinMode(rec, INPUT);
8    pinMode(playe, OUTPUT);
9  }
10
11 void loop() {
12   while (Serial.available()) {
13     data = Serial.read();
14   }
15
16   if (data == '1') {
17     digitalWrite(playe, HIGH);
18     delay(10000);
19     digitalWrite(playe, LOW);
20   }
21   else if (data == '0') {
22     digitalWrite(playe, LOW);
23   }
24 }
```

아두이노와 OpenCV를 연결하는 시리얼 통신 설정

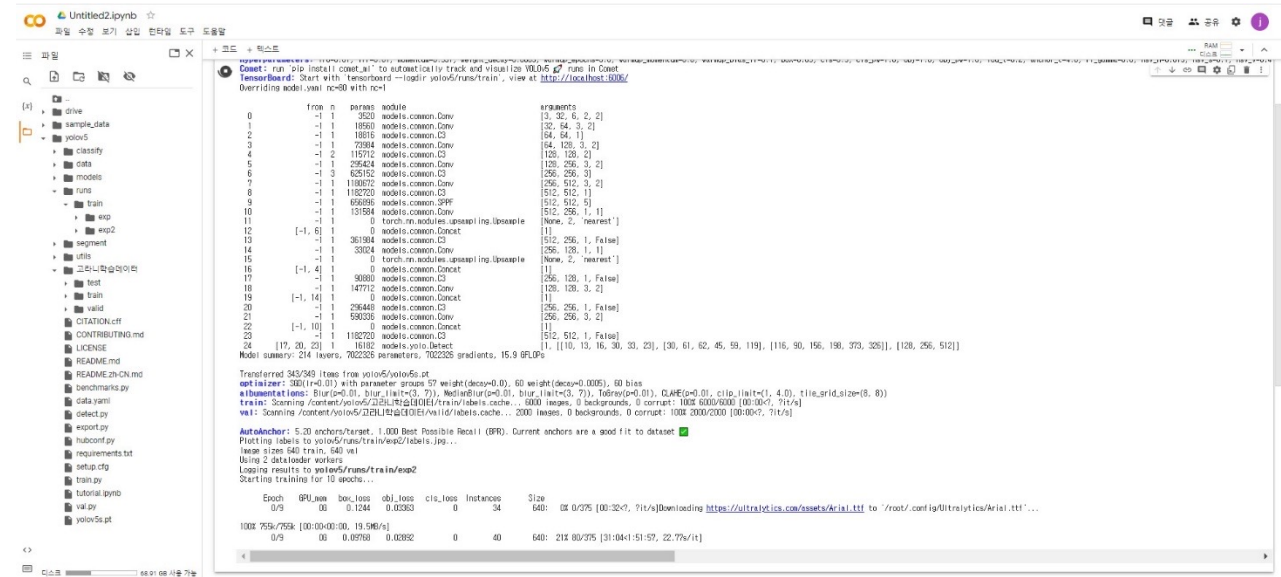
OpenCV로부터 신호가 오면 소리 모듈로 신호를 보내 동작



고양이가 감지 되었을 때 소리 모듈에서 10초 간 소리 출력

# 개발 진행상황

14



- Colab을 통해 YOLOv5에서 고라니를 학습시킴 (train: 6000장, valid: 2000장, test: 2000장)
- 시간이 너무 오래 걸려 중도 멈춤

- 초기 계획했던 YOLOv3, OpenCV를 이용한 고양이 인식 및 침입방지 기능 개발 완료
- YOLOv5, YOLOv8에서 테스트 하였으나 카메라 모듈의 화면을 불러오지 못함
- 추가할 기능 및 아이디어 구상
  1. 고라니 학습 진행 완료 후 객체 탐지 기능 추가 테스트
  2. YOLOv5 상위 버전에서 WebCam으로 객체 인식 테스트
  3. WebCam으로 객체 인식 완료 후 카메라 모듈로 테스트
- 코드 수정 및 보완
- 최종발표 준비

1. 황선규, 『OpenCV 4로 배우는 컴퓨터 비전과 머신 러닝』, (길벗, 2019)
2. 박준원, 『2019한 권으로 끝내는 아두이노 입문 + 실전』, (길벗, 2019)