

OpenCV와 IoT 기술을 이용한 길고양이 퇴치 시스템

정보통신공학과

20181645 신동운

20181675 원석찬

20181688 정재인

지도교수: 이충호

목차

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

연구 배경 및 목적

연구 배경





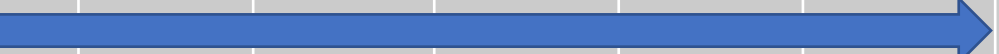
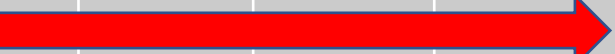


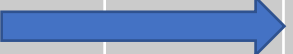


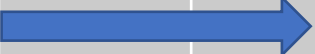
- 길고양이를 퇴치하는 '전기 펜스 설치', '담장 위 장애물 설치', '동작 감지 스프링 쿨러' 등의 방법들은 복잡하고 고비용을 요구함
- 학교 내에 길고양이 퇴치 시스템이 충분하지 않음

연구 배경 및 목적

연구 목적

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

계획대비 진행 상황

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

신동운

- 영상처리 코드 개발
- 인공지능 시스템 개발

원석찬

- 아두이누 모듈 연결
- 아두이노 코드 개발

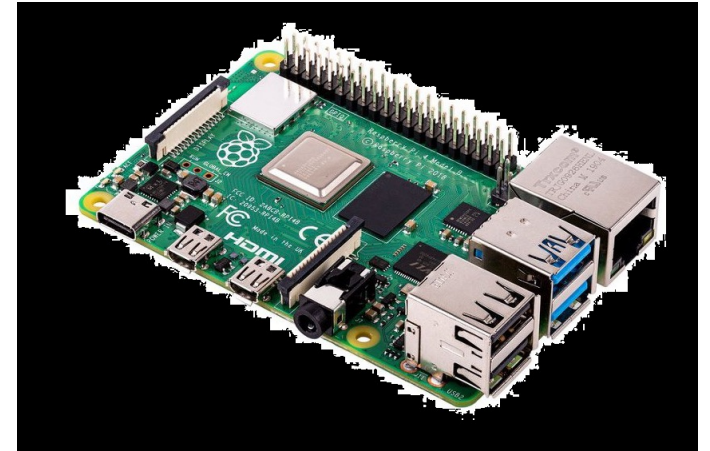
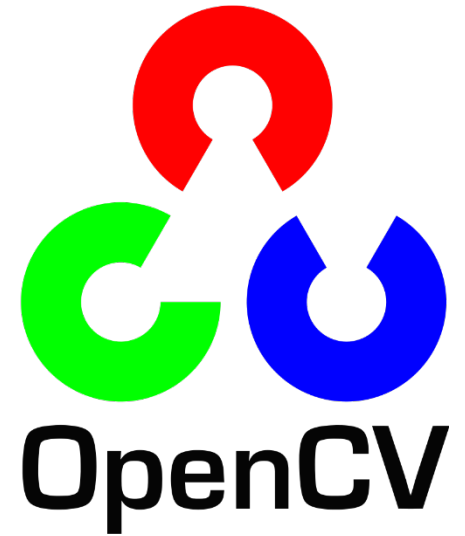
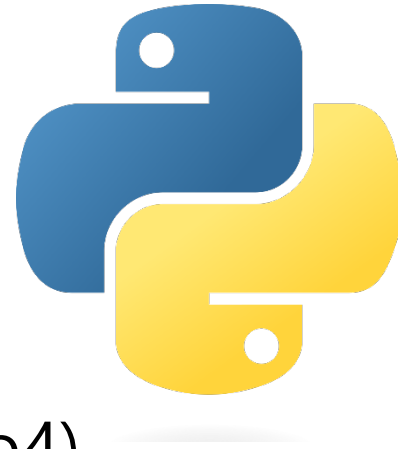
정재인

- 아두이노 모듈 연결
- 영상처리 연구

공통: 보고서 작성, 발표준비

개발 환경

- Python (3.10 ver)
- OpenCV (4.1.2 ver)
- 아두이노 (Arduino IDE v 2.0.3, AVR studio4)
- DESKTOP (Windows 10)
- 카메라 (ESP32 CAM 보드)
- 녹음 모듈



개발 환경



아두이노 키트



녹음 모듈



ESP32 CAM 보드

개발 진행상황

```
import cv2, pafy
```

```
cat_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalcatface.xml')
```

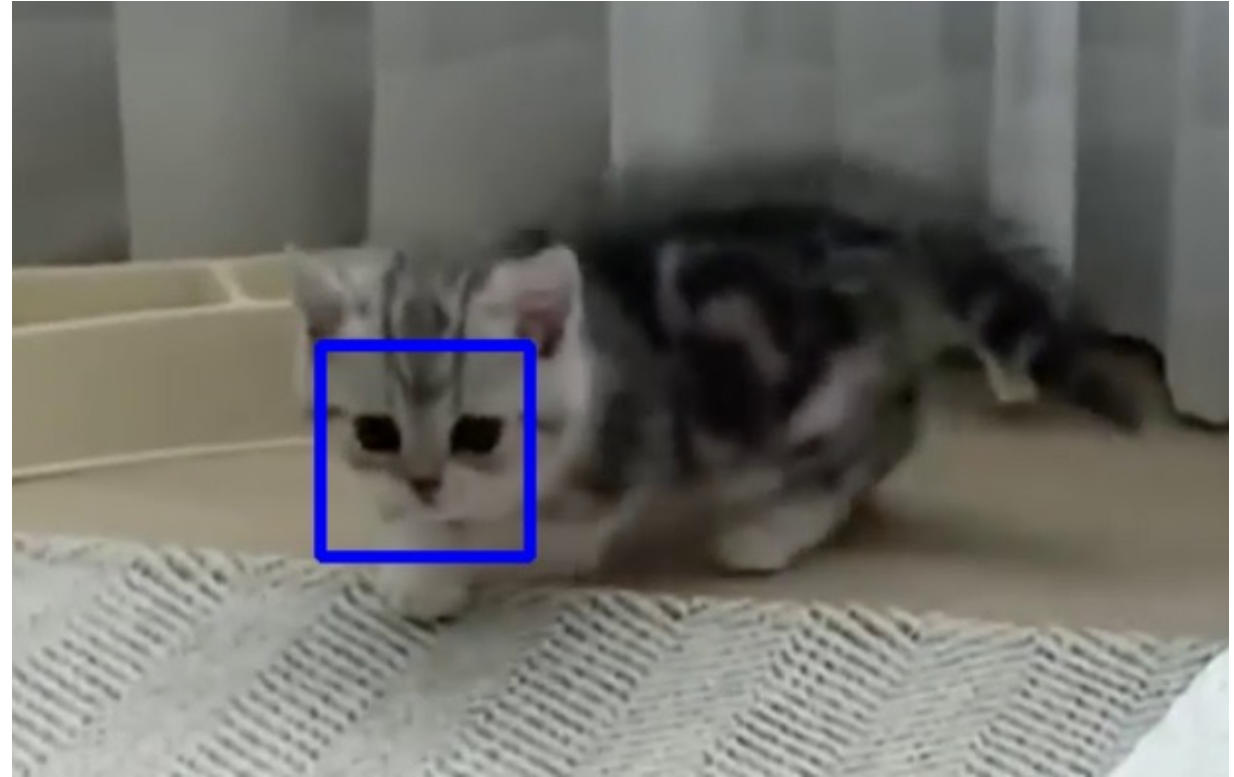
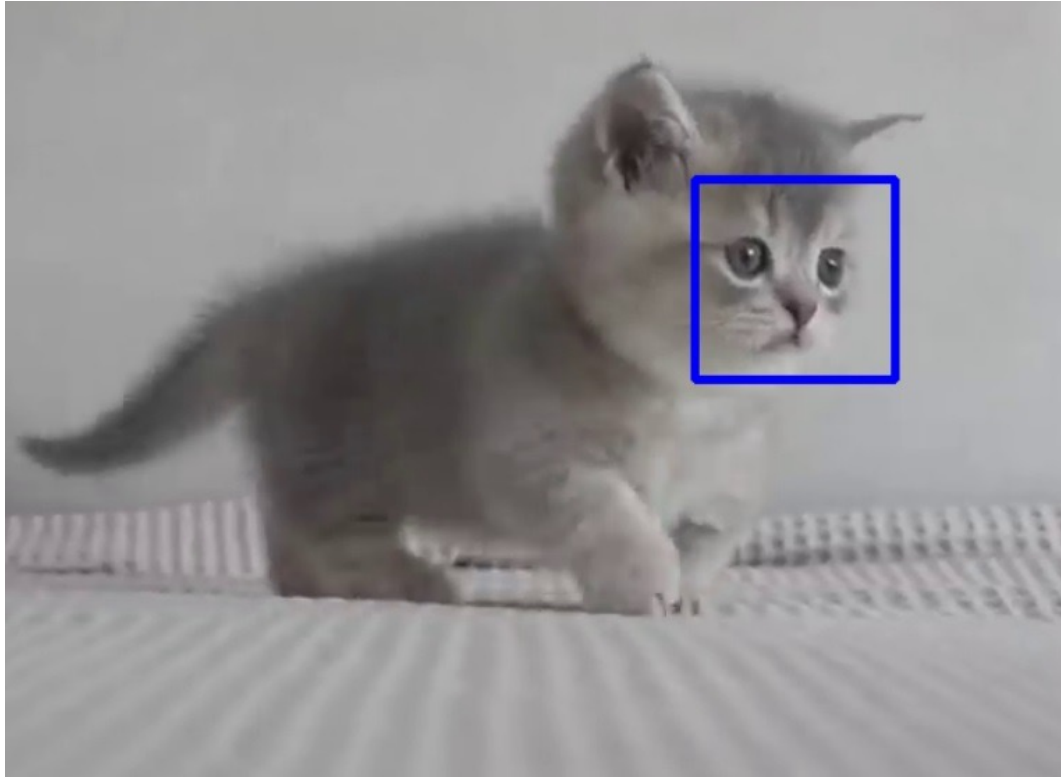
```
url = 'https://www.youtube.com/watch?v=xedmOLcp_EA'  
video = pafy.new(url)  
best_stream = video.getbest()  
cap = cv2.VideoCapture()  
cap.open(best_stream.url)
```

```
while True:  
    ret, frame = cap.read()  
  
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
  
    cats = cat_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))  
  
    for (x, y, w, h) in cats:  
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 3)  
  
    cv2.imshow('Cat Detection', frame)  
  
    if cv2.waitKey(1) & 0xFF == ord('q'):  
        break
```

```
# 리소스 해제  
cap.release()  
cv2.destroyAllWindows()
```

유튜브 영상을 통해 코드 테스트

개발 진행상황



얼굴 인식이 정확하지 않음

개발 진행상황

```
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()
```

- cvlib 라이브러리 추가
- haarcascade 제거

개발 진행상황



좀 더 정확하게 인식하고, 물체의 이름 출력

개발 진행상황

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

```
url = 'https://www.youtube.com/watch?v=VsBdl9tFhFk'
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



다양한 동물도 관측 가능

개발 진행상황

Wifi_ESP32cam.ino

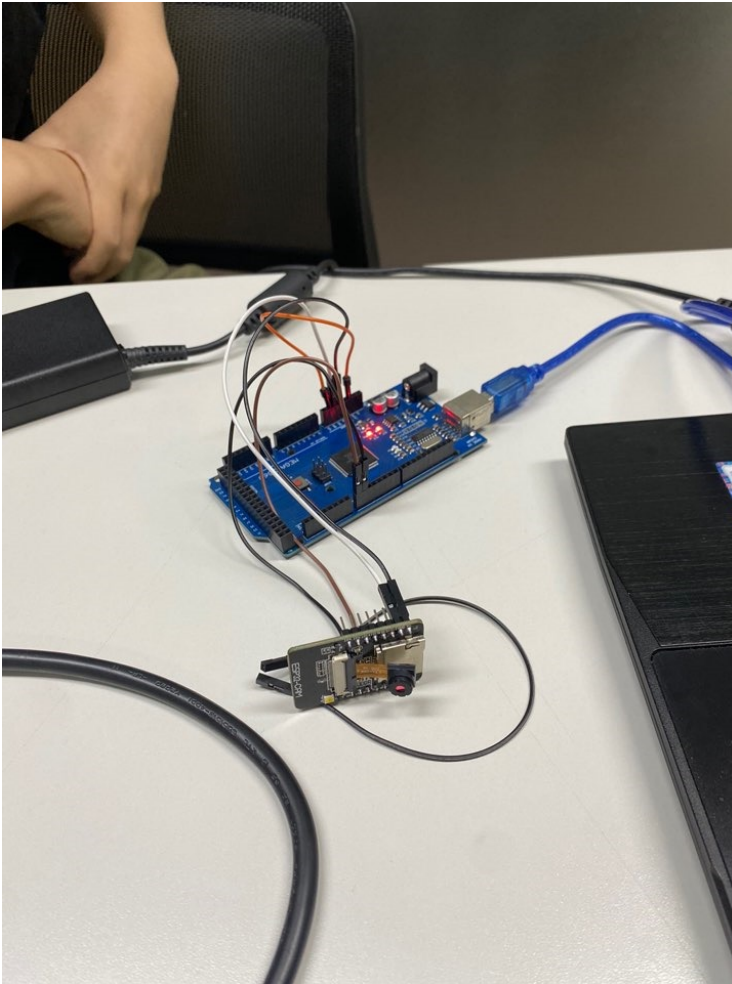
```
1  #include <WebServer.h>
2  #include <WiFi.h>
3  #include <esp32cam.h>
4
5  const char* WIFI_SSID = "HANBAT_LIB";
6  const char* WIFI_PASS = "";
7
8  WebServer server(80); //servidor en el puerto 80
9
10 static auto loRes = esp32cam::Resolution::find(320, 240); //baja resolucio
11 static auto hiRes = 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

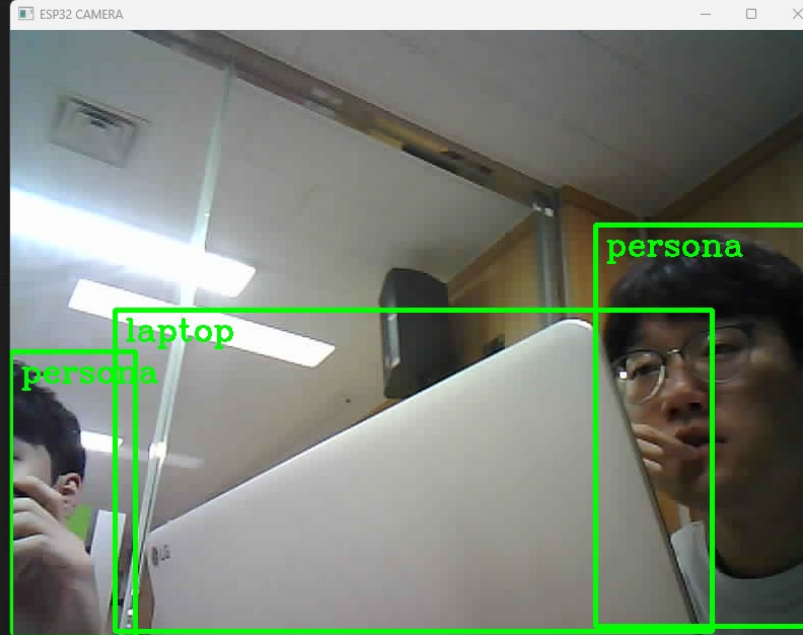


```
streampy x test.py app_httpd.cpp
python > streampy > App > _init_
6
7 class App:
8     def __init__(self, window):
9         #윈도우 생성
10         self.width, self.height = 320, 320
11         self.window = window
12         self.window.geometry("320x320")
13         self.window.title("Read ESP32-CAM")
14         self.window.bind('<key>', self.keyPressed)
15         self.buffer = b''
16         #esp-cap stream
17         url = "http://172.16.130.4:81/stream" #Your url
18         self.stream = urlopen(url)
19
20         self.canvas = Canvas(window, width = self.width, height = self.height)
21         self.canvas.pack()
22         self.delay = 1
23         self.isCapture = 0
24         self.update()
25         self.window.mainloop()
26
27 def keyPressed(self, event):
28     print(event.char)
29     if event.char == 'a': #현재 화면을 사진으로 저장
30         self.isCapture = 1
31
32     if event.char == 'q': #종료하기
33         self.window.destroy()
34
35 def update(self):
36     while True:
37         #받은 데이터 받아오기
38         self.buffer += self.stream.read(2560)
39         head = self.buffer.find(b'\xff\xd0')
40         end = self.buffer.find(b'\xff\xd9')
41         try:
42             if head > -1 and end > -1:
43                 #받은 데이터를 jpg로 변환하기
44                 jpg = self.buffer[head:end+2]
45                 self.buffer = self.buffer[end+2:]
46                 img = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv2.IMREAD_UNCHANGED)
47                 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
48                 img = cv2.resize(img, (320, 320))
49                 # frame = cv2.rotate(frame, cv2.ROTATE_90_CLOCKWISE) # vertical
50                 #사진 화면로 저장하기
51                 if self.isCapture:
52                     cv2.imwrite('capture' + ".jpg", img)
53                     self.isCapture = 0
54                 self.photo = PIL.ImageK.PhotoImage(image=PIL.Image.fromarray(img))
55                 self.canvas.create_image(0, 0, image = self.photo, anchor = NW)
56                 break
57             except:
58                 pass
59         self.window.after(ms=self.delay, func=self.update)
60
61 App().run()
62
```

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

개발 진행상황

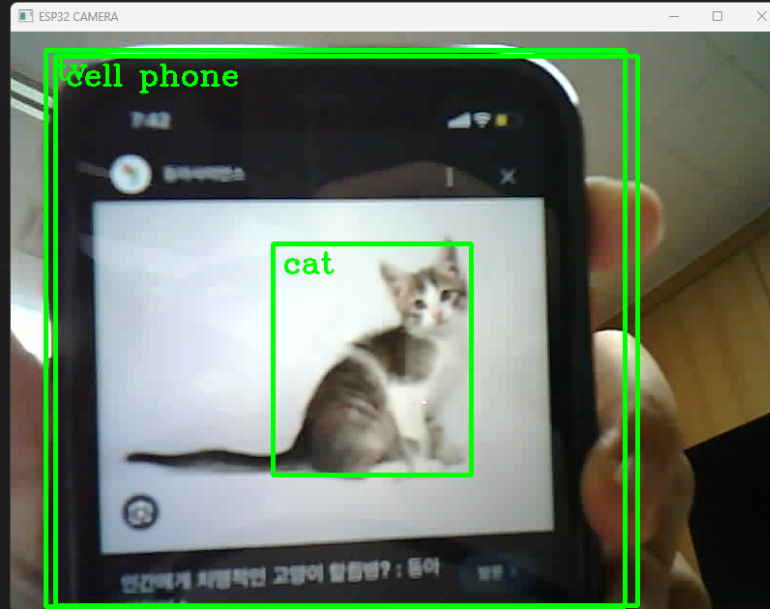
```
python> main.py
5 # python -m pip install opencv-python
6 #참고
7 #https://github.com/Jh0e1RN/ESP32-CAM-wireless-computer-vision-objects-detection
8 #https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/
9 #weights download
10 #https://pjreddie.com/darknet/yolo/
11
12 url = 'http://172.16.130.4/cam-hi.jpg'
13 winName = 'ESP32 CAMERA'
14 # infoPath = './info/'
15
16 classNames = []
17 classFile = r'C:\yoloV3\python\info\coco.names'
18 with open(classFile, 'rt') as f:
19     classNames = f.read().rstrip('\n').split('\n')
20
21 configPath = r'C:\yoloV3\python\info\ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt' #YOLO 환경설정파일
22 weightsPath = r'C:\yoloV3\python\info\frozen_inference_graph.pb' #사전 훈련된 가중치들
23
24
25 cv2.namedWindow(winName, cv2.WINDOW_AUTOSIZE)
26 net = cv2.dnn_DetectionModel(weightsPath, configPath)
27 net.setInputSize(320, 320)
28 #net.setInputSize(480, 480)
29 #net.setInputSize(608, 608)
30 net.setInputScale(1.0/127.5)
31 net.setInputMean((127.5, 127.5, 127.5))
32 net.setInputSwapRB(True)
33
34 while(1):
35     try:
36         #이미지 가져오기
37         imgResponse = urllib.request.urlopen(url) #abrimos el URL
38         imgNp = np.array(bytearray(imgResponse.read()), dtype=np.uint8)
39         img = cv2.imdecode(imgNp, -1) #decodificamos
40         #img = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE) # vertical
41         #img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #black and white
42
43         #사람인식
44         classIds, confs, bbox = net.detect(img, confThreshold=0.5)
45         print(classIds, bbox)
46
47         #사람인식된 경우 박스 및 텍스트 입력
48         if len(classIds) != 0:
49             for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
50                 cv2.rectangle(img, box, color=(0, 255, 0), thickness = 3) #mostramos en rectangulo lo que se encuentra
51                 cv2.putText(img, classNames[classId-1], (box[0]+10, box[1]+30), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
52                 cv2.imshow(winName, img) #mostramos la imagen
53     except:
54         pass
55
56     #ESC key 입력
57     tecla = cv2.waitKey(5) & 0xFF
58     if tecla == 27:
59         break
60 cv2.destroyAllWindows()
61
```



yoloV3를 이용하여 객체인식 확인

개발 진행상황

```
main.py x coco.names
python > main.py ...
5 # python -m pip install opencv-python
6 #참고
7 #https://github.com/jhoelRN/ESP32-CAM-wireless-computer-vision-objects-detection
8 #https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/
9 #weights download
10 #https://pjreddie.com/darknet/yolo/
11
12 url = 'http://172.16.130.4/cam-hi.jpg'
13 winName = 'ESP32 CAMERA'
14 # infoPath = './infor/'
15
16 classNames = []
17 classFile = r'C:\yoloV3\python\infor\coco.names'
18 with open(classFile, 'rt') as f:
19     classNames = f.read().rstrip('\n').split('\n')
20
21 configPath = r'C:\yoloV3\python\infor\ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt' #YOLO 환경설정파일
22 weightsPath = r'C:\yoloV3\python\infor\frozen_inference_graph.pb' #사전 훈련된 가중치들
23
24
25 cv2.namedWindow(winName, cv2.WINDOW_AUTOSIZE)
26 net = cv2.dnn_DetectionModel(weightsPath, configPath)
27 net.setInputSize(320, 320)
28 #net.setInputSize(480, 480)
29 #net.setInputSize(608, 608)
30 net.setInputScale(1.0/127.5)
31 net.setInputMean((127.5, 127.5, 127.5))
32 net.setInputSwapRB(True)
33
34 while(1):
35     try:
36         #이미지 가져오기
37         imgResponse = urllib.request.urlopen(url) #abrimos el URL
38         imgNp = np.array(bytearray(imgResponse.read()), dtype=np.uint8)
39         img = cv2.imdecode(imgNp, -1) #decodificamos
40         #img = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE) # vertical
41         #img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #black and white
42
43         #사물인식
44         classIds, confs, bbox = net.detect(img, confThreshold=0.5)
45         print(classIds, bbox)
46
47         #사물인식된 경우 박스 및 텍스트 입력
48         if len(classIds) != 0:
49             for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
50                 cv2.rectangle(img, box, color=(0, 255, 0), thickness=3) #mostramos en rectangulo lo que se encuentra
51                 cv2.putText(img, classNames[classId-1], (box[0]+10, box[1]+30), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
52                 cv2.imshow(winName, img) # mostramos la imagen
53     except:
54         pass
55
56     #ESC key 입력
57     teca = cv2.waitKey(5) & 0xFF
58     if teca == 27:
59         break
60 cv2.destroyAllWindows()
61
```



yoloV3를 이용하여 고양이 인식 확인

개발 진행상황

```
python > infor > coco.names
1  persona
2  bicycle
3  car
4  motorcycle
5  airplane
6  bus
7  train
8  truck
9  boat
10 traffic light
11 fire hydrant
12 street sign
13 stop sign
14 parking meter
15 bench
16 bird
17 cat
18 dog
19 horse
20 sheep
21 cow
22 elephant
23 bear
24 zebra
25 giraffe
26 hat
27 backpack
28 umbrella
29 shoe
30 eye glasses
31 handbag
32 tie
33 suitcase
34 frisbee
...
```

coco.names 라는 파일에 객체 저장 가능

향후 계획

- 소리 모듈과 아두이노 연결
 - 녹음 및 출력 테스트
- 추가할 기능 및 아이디어 구상
 - 뱀, 고라니 등 다양한 객체 탐지 기능 추가 테스트
- 최종발표 준비
- 코드 수정 및 보완

참고문헌

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