

학습 결과보고서 ①

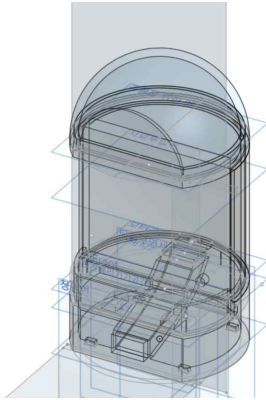
| 프로그램 명 | ①IoT 가전 3D 설계·제작 실무 과정 , ②AIoT 엣지 디바이스 응용 실무 과정 | | | |
|------------|---|---------|-----|------------|
| 참가자 정 보 | 팀 명 | 장하다 | | |
| | 소 속 | AI정보공학과 | | |
| | 성 명 | 장하은 | 학 번 | 2021013230 |
| | 성 명 | 오연서 | 학 번 | 2021013229 |
| | 성 명 | 김경서 | 학 번 | 2022013971 |

학습내용 및 과제결과

사전실습 ①

목표: 펫 기기 외관 또는 부품을 모델링하여 .stl 파일로 완성

<전체 외관 렌더링 이미지>



- 뚜껑, 간식통, 본체 및 바닥, 간식이 나오는 길 등 세부 항목으로 나누어 작업 진행
- 활용툴 : onshape 이용
- 외관 : 원뿔형의 모던한 디자인
- 진행 방식 : 헤드, 바디, 받침, 보호캡 4개의 블록 조립
- 외부에서 LCD확인 가능(전면부 상단에 LCD 부착 공간 마련)

*stl 파일 별도 첨부

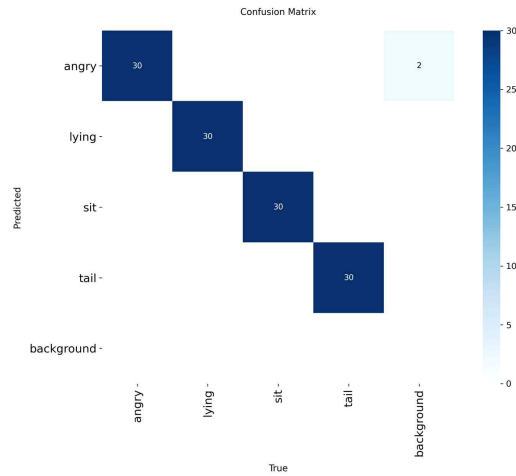
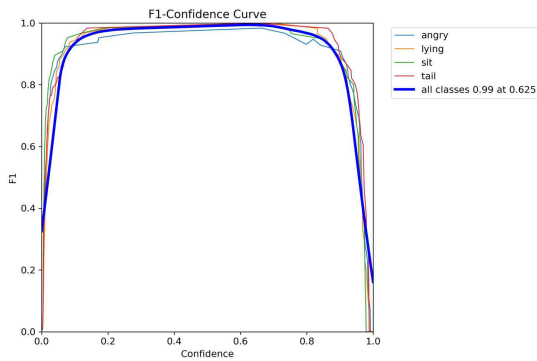
사전실습 ②

목표: 반려동물 얼굴/행동 인식용 데이터 수집 및 AI 모델 학습

- 데이터 수집: 이미지, 4개 분야 30장씩 총 120장
- 라벨링: Bounding Box 라벨링 (labellmg 활용)
- 모델 선택: YOLOv11
- 학습 방법: Pretrained 모델 + 전이학습
- 수집 이미지 : 화난 모습(angry) 30개, 누워있는 모습(lying) 30개, 앉아있는 모습(sit) 30개, 다양한 꼬리모양(tail) 30개 총 120장

- 불안으로 인식하는 경우 : 몸을 낮춤, 몸을 굽음, 앞발들기(상황에 따라) 등
- 정상으로 인식하는 경우 : 몸털기, 걷기, 마운팅, 눕기 등

<성능 평가 결과>



*라벨링 결과 별도 첨부

*학습모델 파일 별도 첨부

*테스트 결과 별도 첨부

사전실습 ③

목표: 라즈베리파이로 얼굴 인식 또는 영상 필터 적용 예제 실행

준비 작업: Raspberry Pi OS 설치, SSH 접속, 카메라 모듈 장착 (완료)

환경 구축: OpenCV 설치 완료

예제 실행: 얼굴 인식 예제 코드 실행

```
GNU nano 7.2 face_detect.py
import cv2

cap = cv2.VideoCapture(0, cv2.CAP_V4L2)
classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

while True:
    ret, frame = cap.read()
    if not ret:
        break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = classifier.detectMultiScale(gray, 1.2, 5)

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    cv2.imshow("Face Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

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

- 실행 시 카메라 화면을 보기 위해서는 ssh 상태가 아닌 원격 화면 접속(VNC)을 통해 라즈베리파이 OS로 실행 필요
- 실시간으로 얼굴 윤곽을 탐지하며, 화면 상에 사각형 박스를 출력하여 인식되고 있음을 표현
- 일정 시간이 지나면 종료 되는 방식이 아닌, q를 누르면 카메라 종료되는 방식

*작동 영상 별도 첨부(실시간으로 영상 속 얼굴 인식하는 모습)



사전실습 ④

목표: Raspberry Pi와 Arduino를 UART, BLE, WiFi 중 하나로 통신 연동

통신 방법 선택: UART 직렬통신

제출물: 연동된 시스템 동작 영상, 코드 스크린샷

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   if (Serial.available()) {  
7     String msg = Serial.readStringUntil('\n');  
8     if (msg == "ping") {  
9       Serial.println("pong");  
10    }  
11  }  
12 }
```

<아두이노 통신 코드>

- ping 이라는 메시지를 보내면 pong 이라는 메시지로 응답하도록 설계

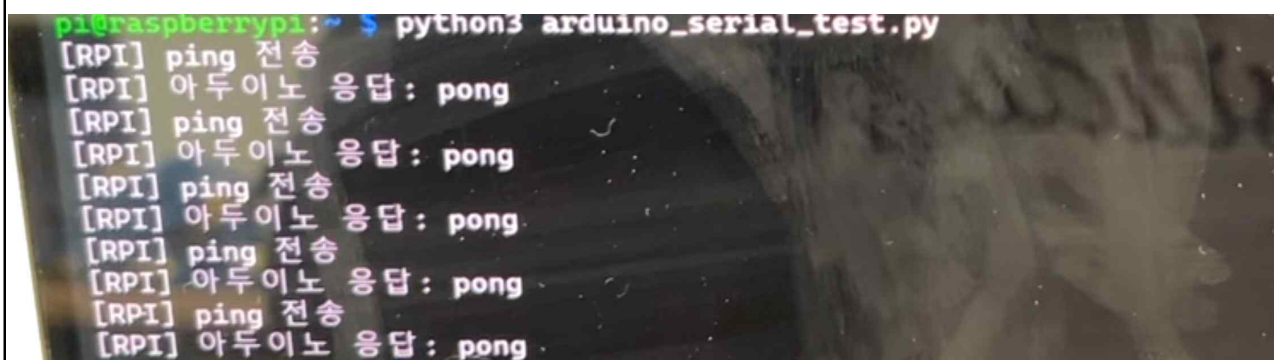
```
GNU nano 7.2      arduino_serial_test.py  
import serial  
import time  
  
# 아두이노와 연결된 포트 (확인한 ttyACM 경로로 변경 가능)  
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)  
time.sleep(2) # 아두이노 초기화 시간  
  
for i in range(5):  
    ser.write(b'ping\n') # ping 메시지 전송  
    print(f'[RPI] ping 전송')  
  
    response = ser.readline().decode().strip()  
    print(f'[RPI] 아두이노 응답: {response}')  
    time.sleep(1)  
  
ser.close()
```

<라즈베리파이 통신 코드>

- 아두이노에서 ping 이라는 메시지를 받으면, pong이라는 메시지로 대답하도록 설계

- Raspberry Pi에서 'ping' 전송 → Arduino가 수신 후 'pong' 전송 → 총 5회 반복하며 시리얼 모니터에 출력

*작동 영상 별도 첨부(5번이라는 정해진 횟수만큼 서로 통신하는 모습)



상기와 같이 특화교육 참가에 따른 학습결과보고서를 제출합니다.

2025년 07월 13일

경상국립대학교 공학교육혁신센터장귀하