**.1. Physical Computing** 혹은 순수 **SW**작품 설명서

## I 주제

**1.** 개발 배경 및 필요성

- ❖ 시력이 좋지않은 노약자나 시각장애인이 약의 성분이나 사용법 등을 제대로 숙지하지 못해 올바르게 복용하지 못하는 경우가 발생한다는 것을 알게되었다.
  - ➢ 이를 해결하기 위해서 의사 혹은 약사에게 직접 상담을 받아야 하는 불편함이 있다다는 것을 깨닫고 개발하게 되었다.
- ❖ 시각장애인, 노인 등 일상 속 불편함을 겪는 사람들이 편리하게 약에 대한 정보를 얻을 수 있다면 약물 오용을 예방할 수 있다고 생각하여 개발하게 되었다.
- ❖ 평소 일반적인 인공지능 서비스로 이용하면서 약품 안내 서비스를 함께 이용함으로써 조금 더 활용할 수 있게 할 것이다.

**2.** 개발 주제**(**목적**):**

- ❖ 노약사, 장애인 등 약물 복용법을 제대로 몰라 약물 복용에 어려움을 느끼는 사람들을 위해 개발했으며, 카메라로 약품을 인식해 사용자에 따라 맞춤형 서비스를 제공하는 작품이다. 이는 시각장애인을 위한 음성인식(STT)과 음성출력(TTS)을 지원하는 서비스이다.
  - ➢ 약품의 복용 횟수, 복용 주기, 알약 개수 및 해당 약품의 부작용 등 다양한 정보를 보다 더 정확하게 안내 하는 것을 목표로 한다.
- ❖ 평상시, 약품 안내 서비스를 사용하지 않을때는 이 기기의 활용성을 높이기 위해 일반적인 인공지능 비서 서빗를 구현하였습니다.

**3.**유사 제품**(**연구**)** 및 차별점**:**

- ❖ 선행 연구 조사 결과, 약품 검색 장치 및 방법이 이미 연구되어있다.
- ❖ 본 작품과 선행 연구와의 차별점
  - 시각장애인과 노인뿐만 아니라 비장애인분들도 사용할 수 있는 범용성있는 서비스
  - ➢ 노약자 및 시각장애인이라는 사용 대상에 맞춰 서비스를 제공한다.
    - TTS(음성 안내):시각 장애인이 보지 못한다는 점
    - 앱이 아닌 기기: 시각장애인이 핸드폰을 사용하기 어렵다는 점
- ❖ 본 제품과 시제품과의 차별점
  - ➢ 일반적인 화면(GUI)을 통한 안내가 아닌 음성을 통한 안내가 이루어진다는 점
  - ➢ 휴대폰을 사용해야 하는 시중 제품과는 다르게 IoT 서비스의 형태로 더 간편하게 사용이 가능하다
  - ➢ .Image Classification을 통한 보다 더 정확한 약품 인식

- ❖ Kipris에서 찾은 유사제품:

  Kipris에서 저희 작품과 유사한 선행 연구를 검색한 결과 완전히 일치한 것은 찾지 못하였으며 '약*로봇'이라 검색한 결과 총 35,365건의 제품이 조회되었다. 그 중 본 작품에도 있는 노인에게 약을 복용하는 시간을 알려주는 알람 기능이 있는 '노인 분들에게 약을 먹는 시간에 맞춰 가져다주는 애견 로봇'이라는 것이 있었다.

## Ⅱ 작품 제작

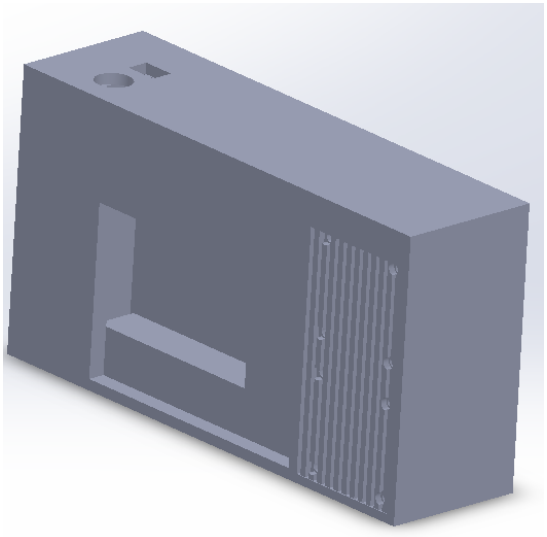
**1.** 작품 설계

- 목표 :
  - ❖ 시력이 좋지 않은 노약자나 시각장애인이 약을 구분하고 약에 대한 성분을 파악하기 힘들어 하는 상황을 해결하기 위함에 초점을 두었다.
  - ❖ 카메라를 통해 해당 약품의 이름과 성분 정보를 인식하면 조회된 내용을 화면과 음성을 통해 안내해주는 서비스다.
  - ❖ 또한 노인 사용자 처럼 건망이 심해 약물 복용 시간을 놓치는 것을 예방하기 위하여 주기적으로 복용해야 약품과 복용시간을 입력하면 알람을 제공한다.

**-** 작동 원리 **:**

- ❖ 평상시에는 시중의 '구글 홈' 같은 인공지능 비서로 사용하다가 '메딕'이라 부른 뒤 원하는 기능을 말하면 해당 기기에서 STT(Speech to text)를 통해 이를 인식하고 기능을 실행한다.
- ❖ "이 약품이 무엇인지 알려줘"라고 하면 기기에 내장된 카메라가 동작하며 해당 약품을 찍으면 기기가 Yolo-V5(Image Classification)을 통해 약을 인식하고 웹 스크랩을 통해 해당 약품 정보를 수집하고 내장되어 있는 정보와 함께 정리해 음성과 화면을 통해 알려준다.
- ❖ TTS를 통해 인식한 약품의 정보를 안내하고 인공지능 서비스를 제공한다

- 작품 설계 **:**
  - ❖ 하드웨어**(H/W)** 설계**:**




- ❖ 솔리드웍스(Solidworks)를 통해 외관을 설계했고 좌측 사진은 전체적인 모습이다.
- ❖ 소리 출력을 위해 스피커를 내장 할 수 있게 되어있다.
- ❖ 상부에는 터치모니터가 장착 되어있다.
  - ➢ 내부에 라즈베리파이(Raspberry-Pi)와 회로가 들어간다.
- ❖ 음성 인식률을 더 높이기 줄 마이크를 사용해 이용하기 편리하게 만들었다.

소프트웨어**(S/W)** 설계**:**

**A. GPIO** 설계**:**

- ➢ 예상 재료**:**
  - Raspberry Pi 3
  - 스피커
  - 소형 마이크
  - 터치 스크린 모니터
  - 카메라

- ➢ Python 언어로 개발한 메인 프로그램과 하드웨어(라즈베리 파이 GPIO) 제어를 실행할 수 있는 Raspberry Pi 3를 사용했다.
- ➢ 정확한 음성인식으로 사용자가 원하는 서비스를 보다 편리하게 제공하기 위해 마이크를 설치했다.
- ➢ 시각장애인을 위한 음성출력(TTS) 서비스를 제공하기 위해 스피커를 내장했으며, 비장애인이나 남녀노소 편리하게 사용할 수 있도록 음성인식 기능을 탑재 터치 스크린 모니터를 사용했고, 해당 기기를 이용한한 전용 프로그램을 개발했다.
- ➢ 또한 이미지나 영상 정보로 약을 인식하기 위해 카메라를 사용했으며, 카메라 모델은 일반 웹캠을 사용했다.

**B. Python:**

시각장애인과 노인이 사용 대상이라는 점을 고려하여 이 분들이 사용하기 편하도록 소프트웨어를 개발하였습니다.

**<**전체적인 알고리즘**>**

- ❖ Raspberry Pi(본체)
  - ➢ Python의 STT(Speech to text)라이브러리를 통해 서비스의 이름(파미)을 인식한다.
  - ➢ 인식한 이름이 맞다면, TTS(Text to speech)를 통해 “무엇을 도와드릴까요?”라고 음성으로 안내
  - ➢ 명령어(즉, 사용자가 필요한 서비스)를 다시 한번 STT(Speech to text)를 통해 입력받는다.
  - ➢ 일반적인 인공지능 비서 서비스인지, 약품 인식 서비스인지 구분하고 그에 따라 작동
    - ■ 일반적인 인공지능 비서 서비스:
      - GPT API를 사용해 입력받은 명령어에 맞는 답변을 제공한다.
    - ■ 약품 인식 서비스:
      - Socket 통신을 통해 “run”이라는 문자열을 pc로 전송해 약품 인식 시작
- ❖ PC(노트북-client1)
  - ➢ Yolo-v5를 통해 이미지 분류를 통해 약품을 구분한다.
  - ➢ 또한번, Socket 통신을 통해 다시 라즈베리파이로 넘어간다.
- ❖ Raspberry Pi(본체)
  - ➢ 해당 약품 정보를 TTS(Text to speech)를 통해 안내한다.

## **1.** 시각장애인을 위한 음성을 통한 안내서비스 개발 **- STT, TTS, GPT API**

(1) 시각장애인이 보지 못하는 점을 고려함과 동시에 평상시 약품 인식 서비스를 활용성을 높이기 위한 음성인식 개발

(2) 음성인식을 통해 서비스를 제공함으로써 점자를 통한 서비스 제공보다 보다 더 범용성을 키움

### **(1)** 음성인식**(STT)**

- 파이썬의 음성인식 모듈인 Speech_Recognition이라는 모듈을 사용했다.
- 실시간으로 음성을 인식하기 위해 분석할 데이터를 마이크에서 실시간으로 가져오게 했다.
- 음성을 text로 바꿔줄 API는 Recognize_google을 사용했다.
- 기본적인 음성 인식을 기반으로 사용자가 서비스의 이름을 부르면 이를 인식하고 이후에 사용자의 음성을 입력받아 원하는 서비스를 실행한다.

❖ 인식한 결과 값에 따른 작동:

```
try:
    while True:
        r = sr.Recognizer()
        with sr.Microphone() as source:
            log_print('recording..')
            audio = r.listen(source,timeout=3)
            print("record complete!")
            try:
                recog_message=r.recognize_google(audio,language='ko-KR')
                log_print('record result : ' + recog_message)
                time.sleep(1)
                if recog_message=="파미야" or recog_message=="밤이야" or recog_message=="바미야" or
recog_message=="타미야":
                    speak("네! 무엇을 도와드릴 까요?")
                    ok=1
                    break
            except sr.UnknownValueError:
                log_print('I can''t understand the speech.')
            except sr.RequestError as e:
                log_print(f'An error has occurred. Causes of the error : error {e}')
```

- 이름 인식:
  - 인식한 결과값이 설정한 이름(파미)가 맞다면 다음 작동인 명령어 인식으로 넘어간다.
  - 인식한 결과값의 약간의 오차는 허용하기 위해 다양한 버전의 이름을 설정해 놓았다.
  - 이름이 맞다면 명령어 인식으로 넘어가기 전 TTS((Text to speech)를 통해 다음 동작 설명
  - try-except 구문을 사용하여 이름이 틀리다면 error 메시지를 보내고 다시 이름 인식을 시도하게 하였습니다.

```
            try:
                recog_message=r.recognize_google(audio,language='ko-KR')
                log_print('record result: '+recog_message)
                diff_result=diff_message("이 약은 무슨 약이야?",recog_message)
                if diff_result== True:
                    socket_message="run"
                    client_socket.send(socket_message.encode())
                    print("메시지 보냄")
                else:
                    message=gpt(recog_message)
                    speak(message)
                    time.sleep(1)
            except sr.UnknownValueError:
                log_print('I can''t understand the speech.')
            except sr.RequestError as e:
                log_print(f'An error has occurred. Causes of the error : error {
    except KeyboardInterrupt:
        pass
```

- 명령어 인식:
  - 일반 인공지능 비서 서비스:
    - Gpt API를 사용해 답변 제공 - TTS
  - 약품 인식 서비스:
    - 약품인식 시작
      - 약품 인식하는 방법은 하단에 기술

**(2)** 음성 출력**(TTS)**

```
def speak(text, lang="ko",speed=False):
    tts = gTTS(text=text,lang=lang)
    tts.save('message.mp3')
    os.system("mpg123 message.mp3")
```

- Python의 GTTS 모듈을 사용했다.
- GTTS를 사용하여 문자를 음성파일인 MP3파일로 변환했다.
- 음성파일을 출력하기 위해 Playsound 모듈을 사용했다.
- 음성 파일을 실행시키기 위해 Linux의 mpg123을 사용하였습니다.
- 기본 인공지능 비서 서비스를 이용할때 긴 답변을 음성으로 변환할 때는 시간이 조금 걸린다.
    - 이를 해결하기 위해 여러가지 대책을 강구하고 있으며 그 대책으로 Linux에서 etts라는 모듈을 테스트해보고 있습니다. 이는 파일로 저장하는 과정이 없이 바로 음성으로 출력해 줍니다.

**(3) Chatgpt API**

```
openai.api_key = "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"
def gpt(i_message:str):
    prompt = i_message
    response = (openai.Completion()).create(
            engine="text-davinci-003",
            prompt=prompt,
            temperature=0,
            max_tokens=300,
            top_p=1,
            frequency_penalty=0.0,
            presence_penalty=0.0,
            best_of=1,)
    gpt_message=response.choices[0].text.strip()
    return gpt_message
```

- OpenAI의 Python 라이브러리인 Openai를 사용했다.
- 유료 버전을 사용하였습니다.
- 엔진은 text-davinci-003를 사용하였습니다.
- token을 300으로 설정하여 더 자세한 답변을 하도록 하였습니다.
- 음성인식한 데이터를 기반으로 TTS로 음성을 출력할때 사용했다.
    - 조금더 수준 높은의 답변을 제공하기위해 사용
        - 조금 더 다양한 말투와 사투리를 인식하기 위함
    - 평상시 일반적인 질문의 답변을 제공하기 위해 사용
- 기본적으로 입력되어있는 명령어와 입력된 명령어가 같은지 비교할때 사용하는 API

```python
def diff_message(message1:str, message2:str):
    gpt_input="\"{}\"와 \"{}\"는 같은 의미야?".format(message1,message2)
    gpt_message=gpt(gpt_input)
    print(gpt_message)
    print(gpt_message[0])
    if gpt_message[0]=='네':
        print("두 말은 의미가 똑같습니다.")
        return True
    elif gpt_message[0]=='아' and gpt_message[1]=='니' and gpt_message[2]=='요':
        print("두 말은 의미가 똑같습니다.")
        return False
    else:
        return "error"
```

- ➢ 기본 형식의 명령어(message1) 파라미터와 입력된 명령어(message2) 파라미터를 입력받는 함수 형식
- ➢ 위의 gpt()함수를 사용해 메시지 비교
- ➢ gpt 메시지의 맨 앞 부분(네/아니오) 대답에 따라 결과값 반환
  1. 명령어가 같음(네) : True
  2. 명령어 다름(아니오) : False
  3. 그 외 경우 및 오류: “error”

## 2. 서버와 클라이언트(YOLO-V5)의 통신

### (1) Socket

- ❖ python의 socket 라이브러리를 사용하여 라즈베리파이와 PC가 통신한다.
- ❖ 통신 로직
  - ➢ 서버와 클라이언트 PC 2대
    - server: PC
    - client: Raspberry Pi 1대, PC1대
  - ➢ IP와 Port를 통해 통신한다.
  - ➢ client에서 데이터를 보내면 server에서 데이터를 받아 다시 모든 client에게 전송하는 방법이다.

**<Server>**

```python
## SERVER ##

import socket
from _thread import *

client_sockets = []

## Server IP and Port ##

HOST = socket.gethostbyname(socket.gethostname())
PORT = 9999

########## processing in thread ##
## new client, new thread ##

def threaded(client_socket, addr):
    print('>> Connected by :', addr[0], ':', addr[1])

    ## process until client disconnect ##
    while True:
        try:
            ## send client if data recieved(echo) ##
            data = client_socket.recv(1024)

            if not data:
                print('>> Disconnected by ' + addr[0], ':', addr[1])
                break

            print('>> Received from ' + addr[0], ':', addr[1], data.decode())

            ## chat to client connecting client ##
            ## chat to client connecting client except person sending message ##
            for client in client_sockets:
                if client != client_socket:
                    client.send(data)
```

```python
        except ConnectionResetError as e:
            print('>> Disconnected by ' + addr[0], ':', addr[1])
            break

    if client_socket in client_sockets:
        client_sockets.remove(client_socket)
        print('remove client list : ', len(client_sockets))

    client_socket.close()

############ Create Socket and Bind ##

print('>> Server Start with ip :', HOST)
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((HOST, PORT))
server_socket.listen()

############ Client Socket Accept ##

try:
    while True:
        print('>> Wait')

        client_socket, addr = server_socket.accept()
        client_sockets.append(client_socket)
        start_new_thread(threaded, (client_socket, addr))
        print("참가자 수 : ", len(client_sockets))
except Exception as e:
    print('에러 : ', e)

finally:
    server_socket.close()
```

연결하기 편하게 하였습니다.

- ❖ Socket을 통해 여러 client한테 데이터를 전송받기 위해 python의 Threading 모듈을 사용하였습니다.
  - ➢ Threading 모듈에 대한 설명은 하단에 기술.
- ❖ Socket의 장점인 장거리 통신이 가능하다는 점에서 기기의 휴대성이 높아진다.

**<Client>**

```python
## CLIENT ##

import socket
from _thread import *

HOST = '' ## server에 출력되는 ip를 입력해주세요 ##
PORT = 9999

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((HOST, PORT))

def recv_data(client_socket):
    while True:
        data = client_socket.recv(1024)
        print("recive : ", repr(data.decode()))

start_new_thread(recv_data, (client_socket,))
print('>> Connect Server')

while True:
    message = input()
    if message == 'quit':
        close_data = message
        break

    client_socket.send(message.encode())

client_socket.close()
```


- ❖ Raspberry Pi(본체)와 컴퓨터에서 실행되어 서버와 통신하는 코드
- ❖ 여기서도 server에서 보내는 데이터를 받기 위해 python의 thread를 사용하였습니다.
- ❖ 본체에서 Yolo-V를 통한 Image Classification을 실행시키기 위해 "run"이라는 데이터를 보낸다.
- ❖ 컴퓨터(client2)에서 인식한 약품 이름을 해당 통신을 통해 본체로 전송한다.

**(2) Threading 모듈**

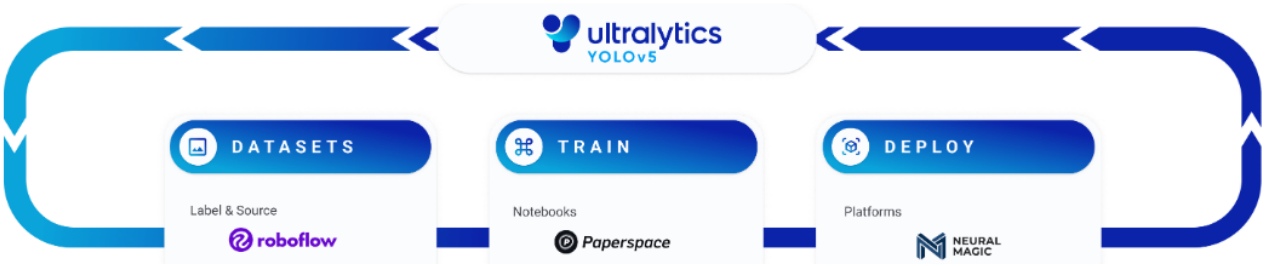
```python
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((HOST, PORT))
def recv_data(client_socket):
    while True:
        data = client_socket.recv(1024)
        data=data.decode()
        print("recive : ",data)

start_new_thread(recv_data, (client_socket,))
```

파이썬 프로그램은 기본적으로 하나의 쓰레드에서 실행된다. 즉, 하나의 메인 쓰레드가 파이썬 코드를 순차적으로 실행한다. 코드를 병렬로 실행하기 위해서는 별도의 쓰레드를 생성해야 하는데, 파이썬에서 쓰레드를 생성하기 위해서는 threading 모듈 혹은 thread 모듈을 사용할 수 있다.

## 3. 카메라를 통한 약품 인식 | Yolo-V5(Image Classification)

### 4)Yolo-V5(Image Classification)



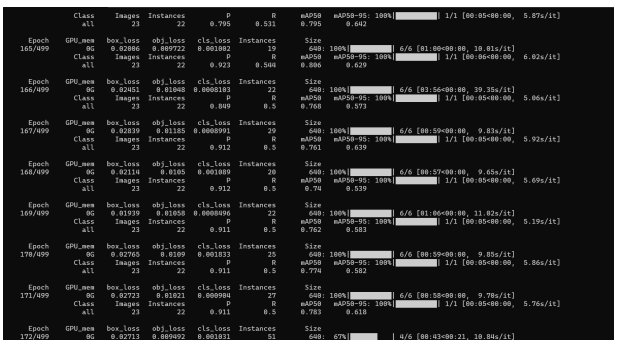

- ❖ 사진에 글자를 인식하는 기술을 OCR(Optical Character Recognition)이라 한다.
- ❖ 본 계획은 python opencv를 통해 OCR을 하여 글자를 추출하는 것이 목적이었으나 텍스트를 추출한 후 약품 이름만 추출하는 방법을 찾지 못하여 이로 대체하였습니다

**(1)** 이미지 학습

**<**라벨링**>**


- ❖ 라벨링을 통해 해당 타이레놀 이 있는 부분을 좌표값으로 지정해주고 class로 분류해 주었다.
- ❖ 약품명 마다 class를 분류해주었으며 해당 class에는 50개의 사진들이 각각 포함 되어있다.
- ❖ 라벨링한 사진을 바탕으로 학습 시킬때 필요한 yaml 파일을 작성하였다.
- ❖ 라벨링한 사진을 바탕으로 train,valid,train 세트를 작성해주었다

**<**이미지 학습**>**

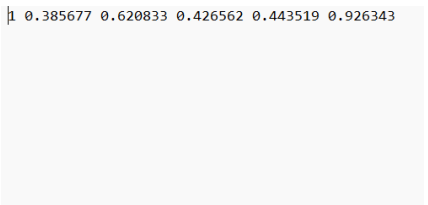


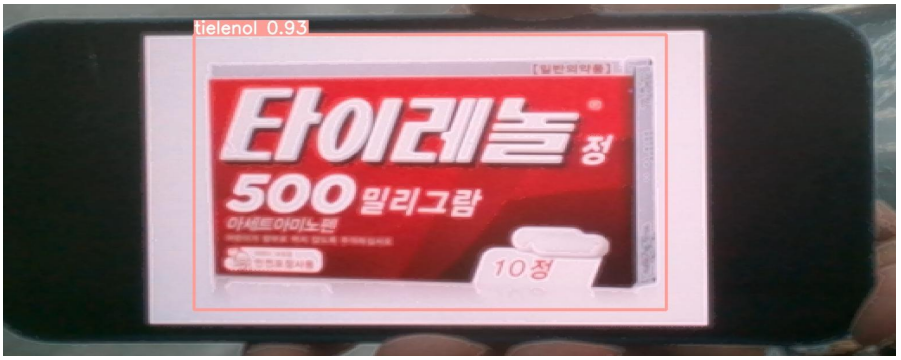
- ❖ 학습 명령어:

  !python train.py --data "data/medicine_recog.ymal" --epoch 500

- ❖ 라벨링을 통해 작성된 이미지와 이미지 데이터 및 yaml 파일을 기반으로 학습된다.
- ❖ 각 약품마다 50장의 사진을 학습시켰으며 에포크는 500으로 하였습니다.

**<**이미지 분류**>**





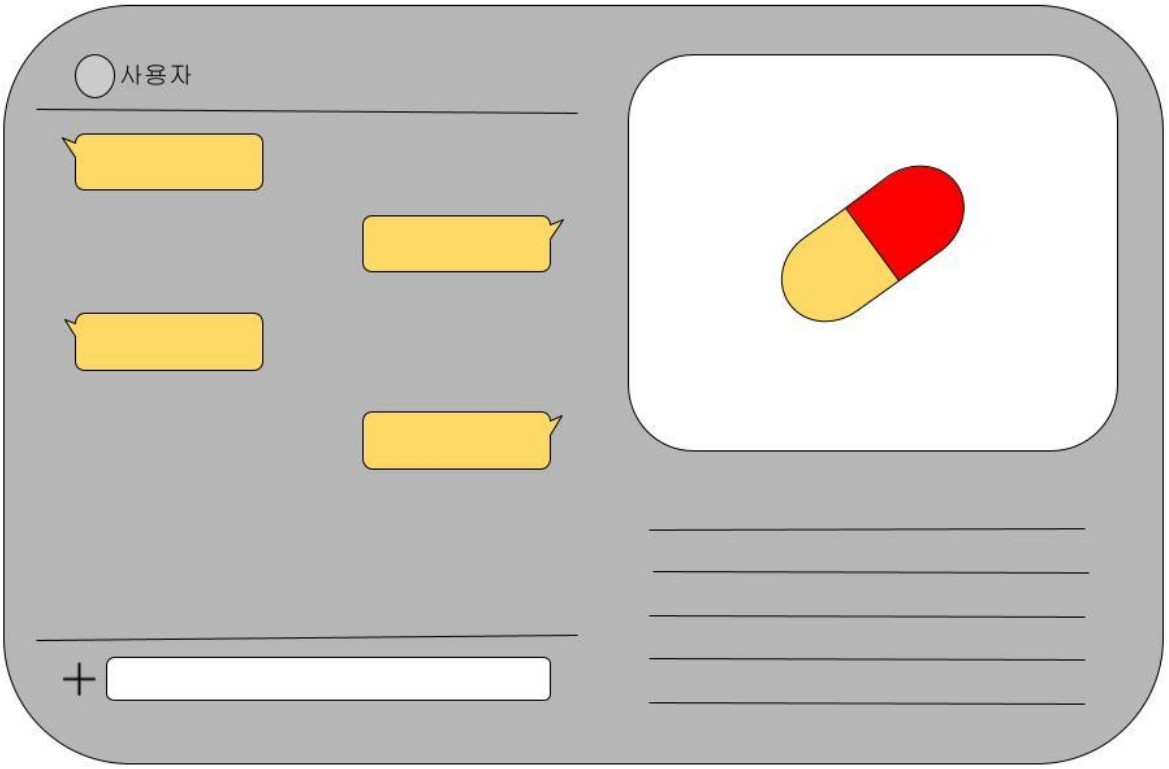

- ❖ 학습된 모델인 best.pt 파일을 이용하여 약품을 인식한다.
- ❖ detect 명령어:

!python C:\yolov5\detect.py --weight "C:/yolov5/runs/train/exp2/weights/best.pt" --source "C:/codepair/save.jpg" --save-txt --save-conf --save-crop

- ❖ —save-txt: 좌표값과 class 번호를 text파일에 저장한다.
- ❖ —save-conf: save-txt에서 만든 text 파일에 confidence(정확도)값을 저장한다.
- ❖ –-save-crop: 인식한 약품의 박스를 잘라 사진을 저장한다.

## **4.** 추후 개발 방향

### **(1)** 웹 개발




- ➢ Html,Css,Javascripts를 사용하여 개발하였습니다.
- ➢ 현재 공부중에 있으며 6월 중순 쯤 개발에 들어갈 예정입니다.
- ➢ A는 음성인식(STT)와 음성입력(TTS)을 사용하여 나눈 대화를 기록하는 창이다.
  - ○ 채팅창 형식으로 표시된다.
- ➢ B는 약품을 인식하는 카메라 화면이 뜨는 창이다.
- ➢ C는 카메라로 인식한 약품의 정보가 뜨는 창이다

**(2)** 크롤링**(Web Crawling)**

```
def crawling():       #크롤링 함수
    url="https://search.naver.com/search.naver?
where=nexearch&sm=top_hty&fbm=1&ie=utf8&query=%EB%85%B8%EC%9B%90%EA%B5%AC+%EB%82%A0%EC%94%A8" #크롤링 웹
페이지 주소
    options = webdriver.ChromeOptions()
    options.add_experimental_option("excludeSwitches", ["enable-logging"])
    options.add_argument("headless")    #크롤링 인터넷 화면 숨기기
    driver = webdriver.Chrome(executable_path="C:\code\WORKSPACE\school\대회\코드페어
\chromedriver.exe",chrome_options=options)#driver  주소 설정
    driver.get(url=url)
    res = driver.page_source
    soup=BeautifulSoup(res,'lxml')

    #날씨 정보
    a=soup.find("p","summary").getText()
    a=a.split(" ")
    ok=0
    weather = a[4]
    # weather="비"
    print(weather)
```

- ❖ 원래 계획은 크롤링을 통해 약품 정보를 수집할 계획이었습니다.
- ❖ python의 beautifulsoup4를 사용하여 HTML 소스에서 데이터를 추출한다.
- ❖ selenium과 web drvier를 사용하여 약품이름을 자동으로 검색할 예정이었습니다.
- ❖ Web drvier의 버전 문제와 라즈베리파이에서의 web drvier 설치 문제로 크롤링을 사용하지 못하였습니다.

**10)**시행착오

- ❖ STT(음성인식)
  - ➢ 음성인식 시 지연이 살짝 있었음
    - ■ 마이크의 문제로 여겨 수차례 마이크 교체
  - ➢ try - except 구문 에러
  - ➢ 이름을 부르고 그 다음 문장을 인식하는 과정 에서 에러가 많이나 계속해서 다시 해야하는 경우가 많음
  - ➢ 음성인식 API를 무료 버전을 사용함으로써 횟수 제한으로 인해 테스트에 제한이 있었음

- ❖ TTS(음성출력)
  - ➢ Pyttts 라이브러리 설치 에러 → git을 사용해 수동 설치
  - ➢ Gtts 버젼 에러(2.3.3 → 2.3.2)
  - ➢ Playsound 모듈 에러(실행이 안됨)

- ❖ Yolo-V5(Image Classification)
  - ➢ labelImg를 설치할때 python과 버전이 맞지 않아 버전에러가 남
  - ➢ pytorch가 python의 버전오류로 설치돼지 않음
- ❖ Chat Gpt Api
  - ➢ 지불 내역이 제대로 인식되지 않아 billing error가 뜸

- ❖ 웹 스크랩
  - ➢ Html에서 text를 추출할 때 <li>,<ul> 태그에서 text만 추출이 안됨
  - ➢ 웹사이트에서 대량의 정보를 추출해오는 과정에서 메모리가 과부하에 걸려 에러가 발생함

- ➢ Beautifulsoup4 & BS4를 설치하고 실행시키는 과정에서 모듈을 인식하지 못함
- ➢ Web Driver를 설치하는 과정에서 Chrome 버전과 맞지 않음
- ➢ 라즈베리파이에서 Web drvier가 설치 되지 않음
  - ■ 이로 인해 웹 스크랩 과정 삭제

❖ Thread

- ➢ Main thread와 서브 thread를 동시에 실행시키지 못함
  - ■ Main thread와 서브 thread가 번갈아가며 실행됨

**2.** 작품 제작

- **4**월 **11**일**(**월**) ~ 4**월 **22**일**(**금**)**
  - **<**아이디어 구상 및 선정**>**
    - ➔ 작품의 대상을 선정(노인 **&** 시각장애인)하고 큰 틀 안에서 어떠한 방법으로 도움을 줄지 논의
    - ➔ 시각장애인을 위한 버스 안내 시스템, 점자 학습기 등 다양한 아이디어 구상
    - ➔ 해당 아이디어들에 대한 기술 실현 가능한지에 대한 조사
    - ➔ 각각의 아이디어들이 어떠한 사회 문제를 해결할 수 있는지에 대한 조사와 논의

- **4**월 **27**일**(**목**) ~ 5**월 **2**일**(**화**)**
  - **<**기술 회의 **&** 외관 구상 및 설계**>**
    - ➔ 각 기능들을 어떠한 방식으로 구현할지에 대한 논의
      - ◆ 라즈베리 파이를 사용하여 구현하기로 함
    - ➔ 라즈베리 파이 보드를 기반으로 외관의 사이즈와 외형을 구상
    - ➔ 구상한 것을 기반으로 솔리드웍스를 사용해 설계하였다.
    - ➔ 모니터를 달기로 함

- **5**월 **8**일**(**월**) ~ 5**월 **10**일**(**수**)**
  - **<TTS** 기술 개발**>**
    - ➔ 약품의 정보를 안내하고 기계의 출력을 담당할 TTS 개발
    - ➔ Playsound, 즉 mp3파일을 실행시켜주는 모듈에 에러가 나 하루 연장
    - ➔ 라즈베리 파이 환경에서의 작동여부 테스트 및 수리
- **5**월 **11**일**(**목**) ~ 5**월 **21**일**(**일**)**
  - **<STT** 기술 개발**>**
    - ➔ 사용자의 명령을 인식할 STT 기술 개발
    - ➔ 해당 기술에서 사용할 파이썬 라이브러리와 API선정
    - ➔ 음성인식 지연 문제
    - ➔ 해당 기술 테스트를 위해 마이크 구매 및 테스트

- 7월 9일(일) ~ 7월 22일(토)
  - **<Yolo-V5** 개발**>**
    - ➔ labelImg를 통한 약품 라벨링
    - ➔ 약품 별 이미지 학습
    - ➔ labelImg 설치 에러 문제 해결
- 7월 23일(일) ~ 7월 29일(토)
  - **<**프로토타입 제작 및 최종 와성**>**

➔ 외관 조립 및 회로 부착
➔ 다시 한번 이미지 라벨링 후 이미지 학습
➔ 음성인식 지연 문제 해결
➔ 프로토타입 테스트 및 작동 영상 촬영

- 추후 개발 방향
  - 카메라 화면과 약품의 정보를 화면에 띄울 수 있도록 web을 개발 할 예정입니다.
  - 음성인식(STT)에서 지연을 해결할것입니다.
  - 외관을 조금 더 수정할 것 입니다.
  - 라즈베리파이에서 yolo를 돌리는 것을 알아보고 있습니다..

## Ⅲ 작품 테스트

❖ 테스트 목적
- ➢ 시각장애인이 이 작품을 사용하면서 불편한점이나 개선해야 할 점이 있는가?
  - ■ 사용자가 음성으로 편하게 제어가 가능한가?
  - ■ 약품을 정확하게 인식하여 사용자에게 정확한 정보를 제공하는가?

❖ 테스트 방법
- ➢ 주제 : 시각장애인에게 약품을 인식하여 정확한 약품에 대한 정보를 제공할 수 있는가?
- ➢ 대상 : 지도교사 및 팀원 2인
- ➢ 테스트 방법: 음성인식을 통해 파미를 호출하고 카메라를 통해 약품을 인식함. 여러가지 약품을 테스트해보며 약품을 제대로 인식하고 보다 더 정확한 정보를 제공하는가? 를 확인함. 이때 공간과 약품에 따라 다양한 결과를 보여줄 것으로 예상해 다양한 환경에서 테스트를 진행해 봄.

❖ 예상 결과
- ➢ 기기가 이름을 인식할때 소음으로 인해 말을 제대로 인식하지 못하고 에러로 처리하는 경우가 있을것으로 예상
- ➢ 이름을 인식하고 다음 명령령어를 인식하는데까지 약간의 지연이 생길 것이라고 예상 함
- ➢ 약품에 따라 인식률의 차이가 조금씩 있을 것으로 예상

❖ 실제 결과
- ➢ 예상한 부분에서 문제가 생겼음
  - ■ 기기가 이름을 인식할때 소음으로 인해 이름을 제대로 인식하지 못해 이상하게 인식하거나 오류로 처리함
  - ■ 말을 인식하기까지 delay가 있음
  - ■ 이름을 인식하고 다음 명령어를 인식하는데까지 delay가 생겨 자연스럽지 못함
  - ■ 이름을 인식하고 명령어를 인식할때 알수없는 이유로 정확도가 떨어짐
- ➢ 이렇게 테스트를 통해 찾은 문제점을 참고해 개발할때 이러한 문제점을 해결하기 위해 여러가지 문제해결방법을 찾음

5. 결과 분석 :
- ➢ 약품에 따라 약간씩의 편차가 있긴하지만 약품 인식 자체에 대한 문제는 없음. 추후에 약품의 학습량과 에포크를 늘려 약품학습을 시켜 약품 별 정화도 편차를 줄일 것임
- ➢ 음성인식 부분에서는 인식률이 저조함. 소음이 있는 환경에서는 정확도가 많이 떨어지며 음성 놈음을 마칠 타임을 제대로 잡지 못하는 경우가 있음

## Ⅳ 해석 **&** 결론

❖ 결론

- ➢ 이와 같은 테스트를 진행함으써 보다 더 정확하게 문제점을 파악하고 해결하여 사용자가 조금 더 편리하고 기기를 사용할 수 있을 것으로 기대한다
- ➢ 일반적으로 시중에서는 시각장애인들이 약을 찾거나 검색하려면 휴대폰의 앱을 통해서 검색해야한다. 이는 휴대성은 더 좋을 수 있어도 시각장애인들이 사용하기에는 불편함이 있습니다. 하지만 저희 작품은 음성서비스를 활용하여 시각장애인들이 보다 더 서비스를 편리하게 사용할 수 있도록 음성으로 상세히 알려줍니다.
- ➢ 또한 메인 서비스인 약품관련 정보만 알려주는 기계로만 쓰다 보면 기기의 활용성이 떨어질 수 있어 조금 더 기기의 활용성을 높이기 위해 평상시 때는 이기기의 장점인 음성 서비스를 활용하여 음성 비서 서비스를 이용 할 수 있습니다.

❖ 추후 개선사항

- ➢ 음성을 인식할때 delay가 생기는현상
- ➢ 음성인식을 할때 소음으로 인해 생기는 에러를 해결하기위해 마이크와 API를 변경해 볼 것입니다.
- ➢ 추후에는 기기를 조금더 가볍게 만들어 휴대성도 좋게 만들것입니다.
- ➢ 일반인도 사용하기 좋게 조금더 기능을 추가시킬 것입니다.