

# **MCI PROGRAM**

## **-TEAM INJEOLMI-**

임지영  
김동률  
박혜빈  
박희현  
백지은  
송승우  
안정은  
정동환

## 목차

### I. 프로젝트 개요

1. 팀원 역할 및 업무	- 1
2. 개발 일정	- 3
3. 개발 환경	- 4
4. 프로젝트 추진 체계	- 5
5. 주제 선정 이유 및 차별점	- 5

### II. 프로그램 설명

1. 비슷한 그림 그리기(유사도 검사)	- 8
2. 글자 색 맞추기(스트루프 검사)	- 11
3. 제시 단어 그리기(글그림 그리기)	- 13
4. 다른 그림 찾기(틀린 그림 찾기)	- 16
5. 숫자 순서 맞추기(기억력 검사)	- 17
6. 문장 따라 말하기(NLP-STT)	- 20

### III. 프로그램 시연

1. 웹 개발	- 22
2. 서버 개발	- 24

### IV. 프로그램 소감

1. 프로그램 확장성	- 28
2. 프로젝트 후기 및 소감	- 30
3. 참고 문헌 / 사이트	- 31

제목	인지 능력 검사 프로그램																																																								
프로젝트 날짜	2022년 12월 26일 ~ 2023년 2월 15일																																																								
팀 이름 / 구성원	인절미 팀 1. 임지영(팀장) : 모델 개발, 웹 개발 2. 김동률(부팀장) : 모델 개발, 웹 개발 3. 박혜빈 : 모델 개발, 웹 개발, 자료 조사, 문서 작성 4. 박희현 : 모델 개발, 웹 개발, 자료 조사 5. 백지은 : 모델 개발, 웹 개발, 자료 조사 6. 송승우 : 모델 개발, 자료 조사 7. 안정은 : 모델 개발, 자료 조사 8. 정동환 : 모델 개발, 자료 조사																																																								
개발 환경	언어 : Python 3.9.7 데이터 분석 : numpy 1.20.3, pandas 1.3.4 OS (운영체제) : Window 11, Window 10 IDE : Colab Python 3.8.16, Visual Code 1.74.0 System : Memory 20G, Hard 256GB 웹 개발 프레임워크 : Flask : 1.1.2																																																								
개발 모델 및 알고리즘	1. 데이터 셋 : 스케치, 아이콘 인식용 다양한 추상 이미지 데이터 사용 (AI Hub) 이후 데이터 추가 2. 알고리즘 : YOLOv5, GAN 3. 사용라이브러리 : numpy 1.20.3, pandas 1.3.4, turtle 4. 사용툴 : Colab, Visual Code																																																								
프로젝트 주요 기능	1. 유사도 검사(객체탐지 이용) 2. 이전에 없던 그림 찾기(GAN) 3. 글자를 통해 해당 그림을 그리기(객체탐지 이용) 4. 단어로 문장 만들기(NLP or DB를 이용하여 문제 입력) 5. 숫자 순서에 따른 기억력 검사(turtle package 이용)																																																								
프로젝트 선정 이유	1. 다양한 사유로 건강증이 찾아지는 요즘, 나의 '인지능력'이 어느정도인지 확인하는 프로그램이 필요해 보임 2. 모든 병이 초기 인지가 중요하기에, 누구나 쉽게 접할 수 있도록 제작 3. '치매'라는 단어보다 '인지 저하'로 범위를 넓혀, 더 많은 사람이 더 빠르고 가볍게 테스트를 받을 수 있도록 함																																																								
개발 일정	<table><thead><tr><th></th><th>1주차</th><th>2주차</th><th>3주차</th><th>4주차</th><th>5주차</th><th>6주차</th><th>7주차</th></tr></thead><tbody><tr><td>주제 선정</td><td>■</td><td>■</td><td>■</td><td></td><td></td><td></td><td></td></tr><tr><td>데이터 전처리</td><td></td><td>■</td><td>■</td><td>■</td><td></td><td></td><td></td></tr><tr><td>데이터 분석</td><td></td><td></td><td>■</td><td>■</td><td>■</td><td></td><td></td></tr><tr><td>모델링</td><td></td><td></td><td>■</td><td>■</td><td>■</td><td>■</td><td></td></tr><tr><td>결과 도출</td><td></td><td></td><td></td><td>■</td><td>■</td><td>■</td><td>■</td></tr><tr><td>web(sann)</td><td></td><td></td><td></td><td></td><td>■</td><td>■</td><td>■</td></tr></tbody></table>		1주차	2주차	3주차	4주차	5주차	6주차	7주차	주제 선정	■	■	■					데이터 전처리		■	■	■				데이터 분석			■	■	■			모델링			■	■	■	■		결과 도출				■	■	■	■	web(sann)					■	■	■
	1주차	2주차	3주차	4주차	5주차	6주차	7주차																																																		
주제 선정	■	■	■																																																						
데이터 전처리		■	■	■																																																					
데이터 분석			■	■	■																																																				
모델링			■	■	■	■																																																			
결과 도출				■	■	■	■																																																		
web(sann)					■	■	■																																																		

## I. 프로젝트 개요

### 1. 팀원 역할 및 업무

#### 1. 임지영 (Back-End)

글자 색 맞추기 : html 에서 결과값을 Flask 로 전달

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

다른 그림 찾기 : 전체적인 코드 작성

문장 따라 말하기 : 만들어진 음성파일 Flask 로 전송

Web\_JavaScript : canvas 기능, 대시보드 구상

Flask : 비슷한 그림 그리기, 글자 색 맞추기, 제시 단어 그리기, 다른 그림 찾기, 문장 따라 말하기  
Flask 연동

DB : 각 게임의 결과값을 DB 에 저장, 쿼리문을 이용하여 결과값을 가져와 점수 치환 후 대시보드  
그래프에 연동

#### 2. 김동률 (Back-End)

숫자 순서 맞추기

- pygbag 명령어를 이용해 기억력게임을 web 에 실행할 수 있도록 구현
- itch.io 인디게임 사이트에 기억력게임 업로드 하여 게임을 flask 로 실행 가능하도록 구현
- EasyOCR : 숫자 순서 맞추기의 결과 이미지를 EasyOCR 을 이용하여 텍스트화

문장 따라 말하기 : STT, NLP API 를 이용하여 모델 구현 및 웹페이지 UI 개선

Flask : 숫자 순서 맞추기, 문장 따라 말하기 기능 Flask 연동

DB : 각 게임의 결과값을 쿼리문을 이용하여 가져오고 대시보드 그래프에 연동

Server : AWS 서버로 프로젝트 파일 업로드 및 실행

Web : 프론트에서 발생하는 여러 오류 수정

#### 3. 박혜빈 (Back-End)

기획 : 전체 프로젝트 구현

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

다른 그림 찾기 : 프로그램 기획 및 시안 제작

문장 따라 말하기 : 프로그램 기획 및 시안 제작, STT, NLP API 를 이용하여 모델 구현

기안서 : 인지 저하 프로그램 기획서 작성

개발서 : 전체 개발서 작성

PPT : 전체 프로젝트 디자인

Web : 웹 About Test, About Us 페이지 제작

회의록 : 전반적인 프로그램 현황 및 노선 작성

#### 4. 박희현 (Front-End)

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

Web\_JavaScript : canvas 기능 및 챗봇 기능 구현&부트스트랩 템플릿 개조 / 다운로드 기능 & 오디오 플레이어 삽입

Web\_CSS : 부트스트랩 템플릿 개조 & 웹 컨셉에 맞는 디자인 작업 진행

UI/UX : 사용자 편의를 위한 전체적인 디자인 구현, 다양한 프로그램을 활용해 웹에 적용

#### 5. 백지은 (Front-End)

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

Web\_JavaScript : 부트스트랩 템플릿 개조, 대시보드 구상 및 구현, 챗봇 기능, 다운로드 기능 및 링크 연결 제공

Web\_CSS : 부트스트랩 템플릿 개조, 웹 컨셉에 맞는 디자인 작업, 대시보드 정돈 및 구현

UI/UX : 웹 전체적인 로드 맵 구상 및 구현, 사용자 편의를 위한 전체적인 디자인 배치, 다양한 프로그램을 활용해 웹에 적용

Design : 팀 로고

기획 : 주제 선정 및 데이터 확보

개발서 : 개발취지 작성

#### 6. 송승우 (Back-End)

비슷한 그림 그리기 : VGG16 모델 활용하여 이미지간 유사도 판별 후 Preprocessing 데이터 증강 숫자 순서 맞추기

- EasyOCR : 숫자 순서 맞추기의 결과 이미지를 EasyOCR 을 이용하여 텍스트화

문장 따라 말하기

- STT : 녹음한 음성 파일 텍스트로 변동
- NLP : 녹음한 음성 파일이 동등한 의미인지 판별
- TTS : 제시된 텍스트를 음성으로 재생

Flask : 문장 따라 말하기 기능 Flask 연동

Server : AWS S3 사용, AWS LightSail, Nginx 및 MobaXterm 활용

DB : 게임, Back-End, Front-End 연동 및 SQL 사용

#### 7. 안정은 (Back-End)

기획 : 전체 프로젝트 스토리보드 작성

비슷한 그림 그리기 : VGG16 모델 구축

글자 색 맞추기 : 전체적인 코드 작성

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

숫자 순서 맞추기 : 전체적인 코드 작성

문장 따라 말하기

- STT : Etri API 사용

개발서 : 모델별 로드맵 개발서 작성

PPT : 전체 프로젝트 디자인

Web : 웹 페이지 배경 디자인

#### 8. 정동환 (Back-End)

비슷한 그림 그리기 : VGG16 모델 활용하여 이미지별 feature 추출용으로 사용

글자 색 맞추기 : 백엔드 최적화

제시 단어 그리기

- Roboflow : 이미지 라벨링 및 증식
- YOLOv5 : 전처리 된 이미지를 사용하여 정확도 상승

다른 그림 찾기 : 백엔드 최적화

문장 따라 말하기 : STT, TTS, 문장 유사도 검출용 Api 찾기

Flask : 기존 app.py 파일 대신 blueprint 적용

Server : AWS S3 사용, Nginx 및 Gunicorn 활용

Web : 백엔드 request 관련 최적화 DB

Error : 온갖 Error 고치기 도전 경험 다수

## 2. 개발 일정

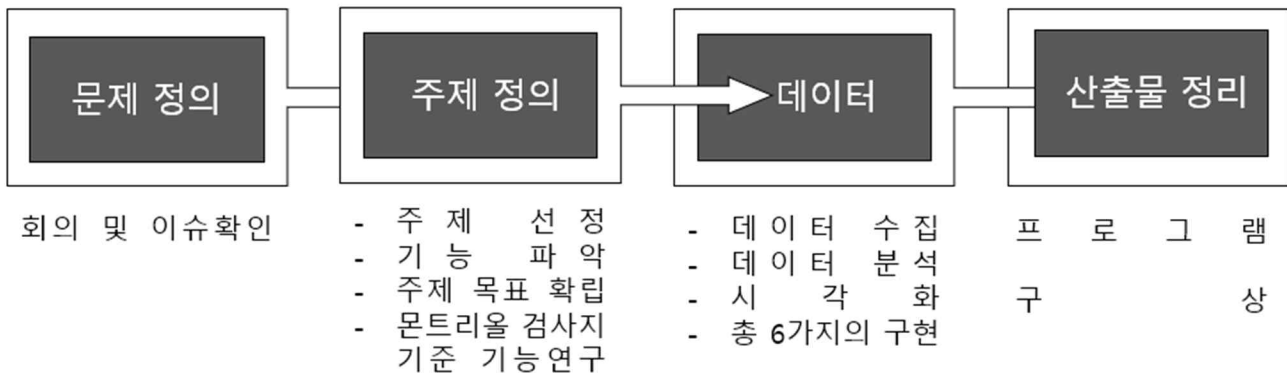
월		12월	2023년 1월					2023년 2월		
일정		5주차	1주차	2주차	3주차	4주차	5주차	1주차	2주차	3주차
프로젝트 계획	주제 선정	12월 26일 ~ 1월 5일								
	기안서 작성	12월 27일 ~ 1월 11일								
자료 수집	모델 기준 확립	12월 26일 ~ 1월 11일								
기능구현	그림 그리기		프로그램 구축 및 플라스크 부착							
	기억력 게임		프로그램 구축 및 플라스크 부착							
	문장 따라 말하기		프로그램 구축 및 플라스크 부착							
	스트루프 게임		프로그램 구축 및 플라스크 부착							
	유사도 검사		프로그램 구축 및 플라스크 부착							
	틀린 그림 찾기		프로그램 구축 및 플라스크 부착							
웹 구현	프론트 웹				1월 16일 ~ 2월 14일					
	백엔드 웹				1월 10일 ~ 2월 14일					
자료 분석	자료 수집	12월 26일 ~ 1월 24일								
	자료 분석		1월 2일 ~ 1월 12일							
평가 및 전개	개발서 작성				1월 17일 ~ 2월 14일					
	PPT 작성					1월 24일 ~ 2월 14일				
	발표 준비									
	리허설									
	발표									2/15

## 3. 개발 환경

- Language : Python 3.9.16, Python 3.10, HTML5, CSS3, JavaScript
- Analysis Tool : numpy 1.20.3, pandas 1.3.4
- OS : Window 11, Window 10
- IDE : Colab Python 3.8.16, Visual Code 1.74.3
- System : Memory 20G, Hard 256GB
- DBMS : SQLite
- Flame work for WEB : Flask 1.1.2
- 
- Model : STT, TTS, NLP(ETRI API), Yolov5, VGG16
- Other Tools : Psychopy, itch.io( 추후 각 기능별로 사용하는 개발 환경은 프로그램 설명 파트 내에서 추가적으로 설명할 예정)

#### 4. 프로젝트 추진 체계

● WBS(Work Breakdown Structure) :



주제 정의를 위해 우선 주제를 선택한 다음, 기능을 파악하고자 했고 이후 확실한 주제 목표를 확립하는 단계에까지 도달할 수 있었다. 다음으로 데이터 준비 및 탐색을 위해 몬트리올 검사지를 기준으로 기능을 연구하게 되었고, 데이터 분석 및 검증을 통해 총 6 가지의 기능을 구현하고자 한다. 이를 통해 6 가지의 기능이 담긴 웹을 구현해 최종 프로그램을 만드는 것을 목표로 잡았다.

#### 5. 주제 선정 이유 및 차별점

최종 프로젝트를 준비하게 되면서 지금까지 실행했던 알고리즘과 모델들에 대한 여러 관심을 쏟아볼 수 있었다. python 과 R, Machine Learning 과 Deep Learning 까지 학습했고, 그 중에서 특정한 학습을 지정하는 것보다 모든 과정의 내용이 담겼으면 하는 기대를 통해 프로젝트를 기획하게 되었다.

가장 먼저 주제를 찾기 위해 주변에 보이는 사물에 관해 관심을 돌려보고, 현시점에서 보이는 뉴스 등을 통해 키워드를 찾아보게 되었다. 이에 어린이집보다 양로원의 시설이 늘어나고 있다는 소식을 통해 '고령화'에 대해 고민하기 시작했다.

하지만, 노인에 대한 소식은 너무나 포화 상태였기에 쉽사리 선택하지 못했다. 그렇다면, 고령화에 집중하기보다 '전 연령대'로 시선을 돌려보는 건 어떨까 하는 생각을 통해 '인지 능력 저하'로 설정하게 되었다.

우리는 이에 관한 데이터를 수집하고, 통계청에서 제공하는 보건복지부 자료로 2020 년부터 2022 년까지 전국 연도별 인지 최경도 환자, 경도 환자, 중증도 환자, 중증환자 수와 전국 연도 및 연령별 경도 인지 장애 환자수를 확인하였다.

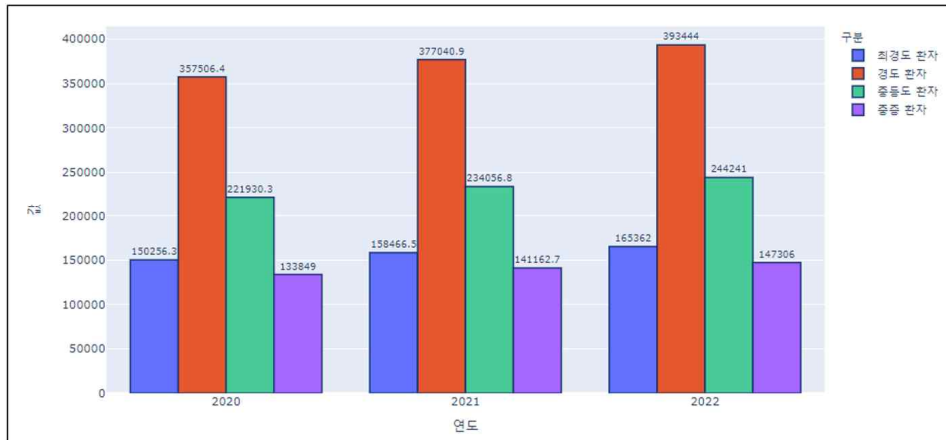


그림1) 2020년~2022년 인지저하인구

(출처 : 2022 보건복지부)

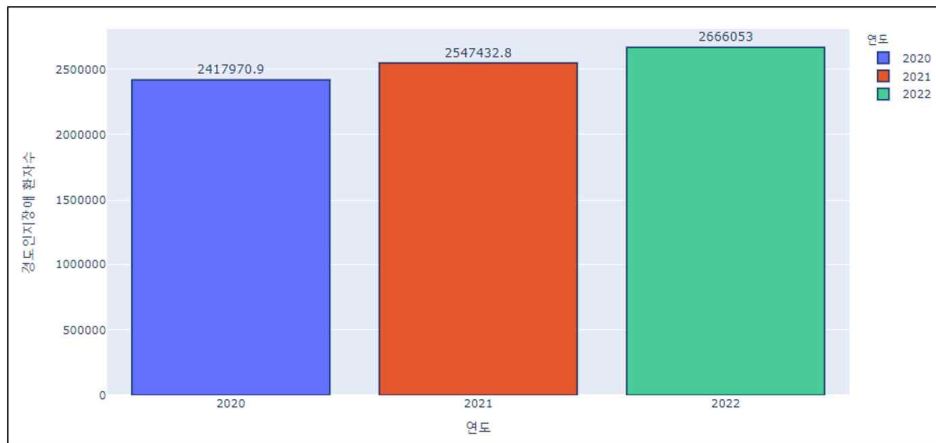


#### 전국 연도별 최경도환자, 경도환자, 중증도 환자, 중증환자 수



분석1. KOSIS 2022보건복지부

#### 전국 연도별 경도인지장애 환자수



분석2. KOSIS 2022보건복지부

해당 그래프(분석 1, 2)를 보면 알 수 있듯이 매년 꾸준히 인지 저하를 겪고 있는 인구가 늘어나는 것을 볼 수 있다. 이제 인지 저하는 노년에서만 겪고 있는 문제가 아닌, 전연령대가 겪고 있는 고민이라는 생각을 할 수 있다.

위의 그래프(분석 3)를 보면 알 수 있듯이 경도 인지 장애(Mild Cognitive Impairment, MCI)에 조기로 발견할 시 늦게 치료했을 때 보다 치매를 늦출 수 있다고 보인다. 순천향대학교 구미병원 신경과 안병준 교수는 "경도인지장애는 알츠하이머형 치매로 이환 될 확률이 정상군에 비해 높으므로 이들의 인지기능 변화에 대한 면밀한 경과 관찰이 필요하다"며, "경도인지장애와 같은 인지 기능의 저하를 빨리 알아차리고 예방적 차원에서의 관리를 시작한다면 치매로의 이환을 막거나 늦출 수 있다"고 설명했다.

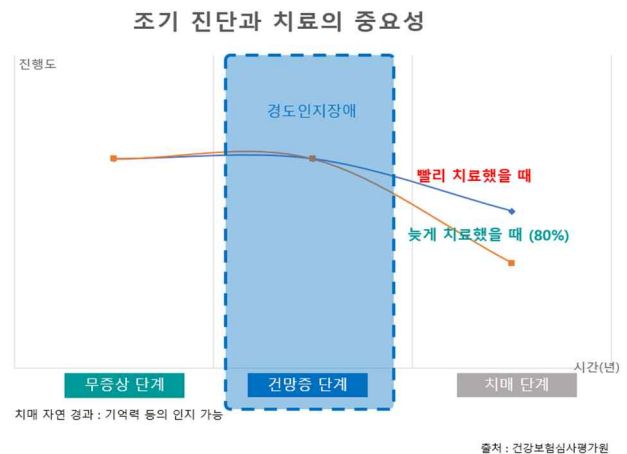
이에 '인지 능력 저하'라는 주제 선정 이유와 차별점을 설명하고자 한다.

### <주제 선정 이유>

- ① 다양한 환경과 사유들로 인해 건망증이 찾아지는 요즘, 나의 '인지 능력'이 어느 정도인지 확인하는 프로그램이 필요해 보인다 여겨졌다. 단순히 하려던 이야기를 까먹고, 물건의 위치를 잠깐 잊어버릴 때면, '내가 혹시...?' 라는 생각을 하는 사람들이 늘어나고 있기 때문이다.
- ② 모든 병은 초기부터 '인지 저하'를 확인하는 것이 중요해 누구나 쉽게 접할 수 있도록 제작하는 것이 필요하다고 생각된다. 질병이 발병하기 전부터 예방하는 일도 필요하지만, 자기도 모르는 사이에 발병한 질병을 인지하지 못하고 있다가 점점 악화되는 것을 막는 것 또한 무시할 수 없다고 생각했다.
- ③ '치매'라는 단어보다 '인지 저하'로 범위를 넓혀, 더 많은 사람이 더 빠르고 가볍게 테스트를 받을 수 있도록 한다. 단순히 '치매'라는 말을 듣게 될 때, 대부분은 질병을 떠올려 겁을 내거나 자신의 이야기는 아니라 생각하는 경우가 대다수이다. 그래서 우리는 '인지 저하'로 범위를 넓혀 병증에 이르지 않더라도 누구나 겪어봤을 경험에 초점을 맞췄다. 사용자의 경각심을 일깨우고, 본 프로그램의 접근성을 높이는 효과를 기대한다.

### <다른 프로그램과의 차별점>

- ① 기존의 인지 저하 프로그램은 대부분이 노년층만을 대상으로 하고 있다. 과거에는 기억력에 대한 문제들이 노년층에서만 나타난다고 생각했기 때문이다. 하지만, 연령대는 계속해서 변화하기 시작했고, 기준이 되었던 노년층에서 벗어나 빠르게는 20 대에도 찾아오는 문제들이 존재해 프로그램의 제작이 필요하다고 생각했다. 노년층만이 아닌 기억 인지를 걱정하는 전 연령을 대상으로 한 프로그램을 제작하고 싶어 차별점을 두는 것을 계획한다.



우선, 차별점을 설명하기 전 알아야 할 개념이 존재한다. 스타트업의 비즈니스 모델을 일컫는 개념에는 '페인킬러(pain-killer)'와 '비타민(Vitamin)'이 있다. 전자는 문제를 해결하는 것에, 후자는 활력소의 역할을 하는 것에 방점이 찍혀 있다. 이를 통해 다음 차별점을 설명하고자 한다.

- ② 대부분의 검사들은 비타민 요소만 들어간 프로그램이거나 페인킬러 요소만 들어간 프로그램이라 구분 짓고 있다. 기존 병원이나 상담 내에서 실행하는 검사는 정확하지만, 딱딱한 테스트로 끝나는 과정이 대부분이다. 이에 우리는 정적인 검사가 아닌 조금 더 쉽게 접근할 수 있으면서도 현장에서 사용되고 있는 몬트리올 검사에 기초하여 신뢰도를 잃지 않으려 노력해 비타민 요소와 페인 킬러가 적절하게 섞인 프로그램을 만들고자 한다.
- ③ 현재에는 기억 인지나 건망증을 대수롭지 않게 생각하는 사람들이 많다. 실제로도 경각심을 느끼지 않고 남에게 생긴 일이라 생각하는 사람들이 많이 존재한다. 이러한 사람들에게 병원에 가기 겁이 난다면, 혹은 추후에 일어날지도 모르는 일을 미리 예방하고 싶다는 생각이 들 때 권유할 수 있는 프로그램을 만들고자 한다. 본 프로그램을 통해 딱딱하게 느껴지던 정신건강학에 관심을 갖게 되고, 생활 속에서도 인지 능력에 대한 자각과 함께 건강한 삶에 기여할 수 있기 바란다.

## 2. 프로그램 설명



### 1) 비슷한 그림 그리기

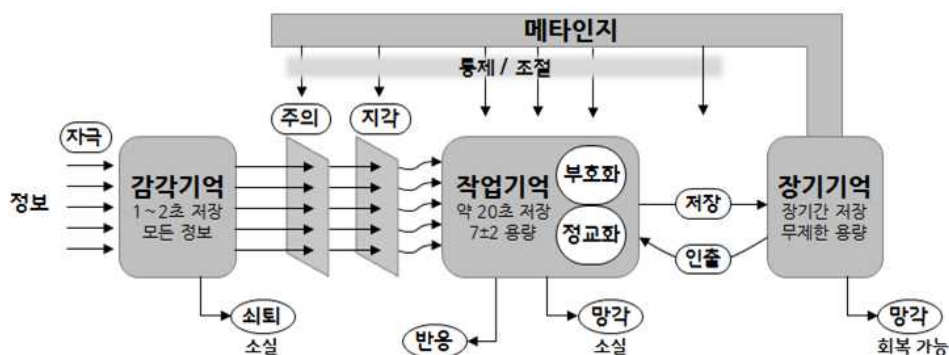
: 해당 단어에 대한 인지를 가지고 있음 동시에 사물의 모습을 기억해 구현

#### (1) 개발 취지

미국 알츠하이머 협회(Alzheimer's Association)의 루스 드류(Ruth Drew)는 미술을 포함한 예술 활동이 인지 경도를 겪고 있는 사람에게 지금보다 더 좋은 날('good moments, good hours and good days')을 사는데 도움이 되며 예술 활동을 할 때 불안감이나 우울증이 완화된다고 말한다.

하지만 국내에서는 치매 환자의 미술 활동이 그리 활성화되고 있지 않은 실정이다. 치매미술치료의 효과가 어느 정도 인지 잘 알려지지 않을 탓이다. 1991 년 치매 미술 협회를 설립한 이후 지금까지 치매미술치료를 알리며 수많은 치매 노인들을 돌봐 온 신현옥 협회장에게 그림 그리기 효과에 대해 물었다. 그녀는 "그림을 그리는 행위가 뇌를 자극해 치매 예방을 돕는다고 말하기에는 그 이상의 것을 내포하고 있다."고 말한다.

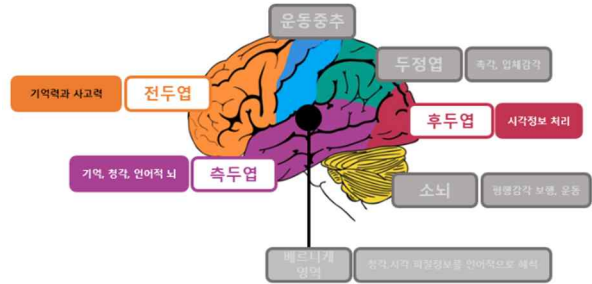
(출처 : 라이나 생명 -전성기\_매거진 中)



우리는 위의 기사와 더불어 K-MOCA(한국형 몬트리올 인지평가)를 기반으로 프로그램을 구성했다. 해당 프로그램은, 이미지를 일정시간 보여준 뒤(감각기억에서 단기기억으로 외부자극을 저장) 해당 그림을 기억하여 그리는 것(단기기억을 다시 도출해내는 작업)을 제안한다.

#### (2) 해당 검사의 기대효과 :

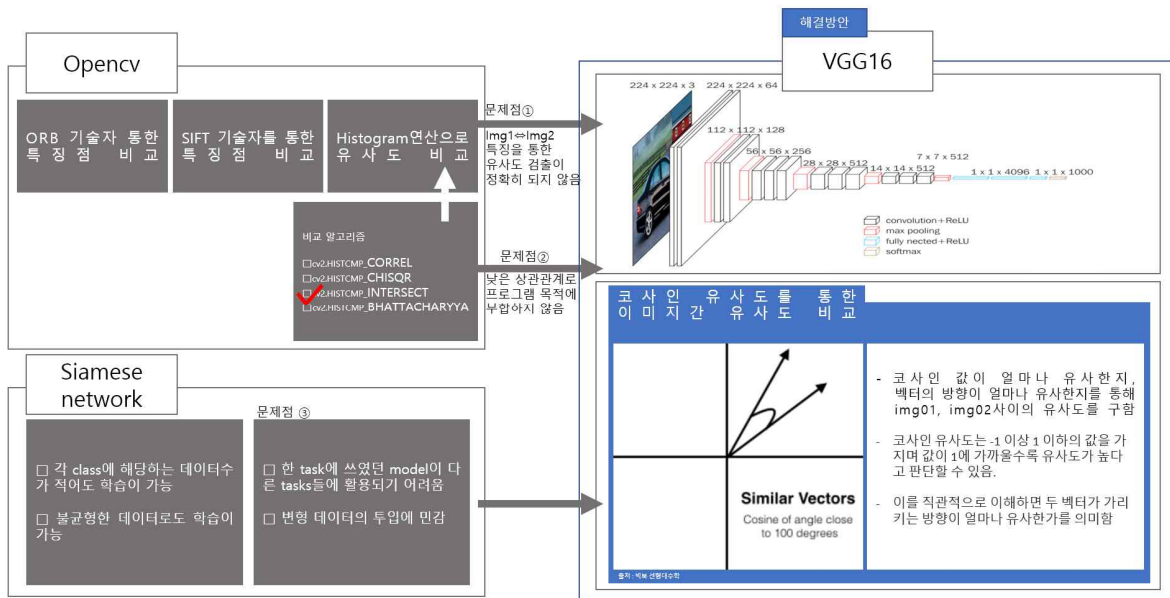
- ① 시각적 학습
- ② 기하학적 모양에 대한 기억력 및 재인
- ③ 즉각적인 및 지연된 상기
- ④ 소근육 조절



### (3) 개발

#### 가 개발 작업 :

- 문제점 및 해결방법(모델 구현도)

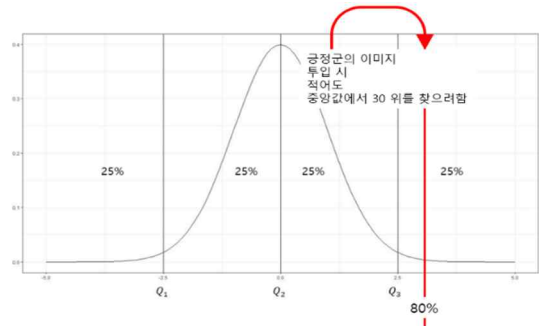


첫번째 접근으로는 OpenCV 를 이용했다. 기존 이미지와 검사 이미지의 유사도를 구하기 위해 OpenCV 를 사용하여 두 이미지(기존 이미지와 판별 이미지)간의 유사도를 판별하려 시도했다. OpenCV 알고리즘 중 ORB 기술자를 통한 특징점 비교, SIFT 기술자를 통한 특징점 비교하여 이미지에서 특정 특징을 추출하 다음, 특징 간의 거리를 계산하여 두 이미지의 유사도를 판별하려 했으나, 손으로 그리는 이미지 특성상, 풀링(Pooling)을 하는 과정에서 많이 깨지기 때문인지 결과값이 좋지 않았다. 그리고 Histogram 연산을 통해 두 이미지의 히스토그램(이미지의 각 픽셀 값의 분포)를 계산한 다음, 히스토 유사도 지표 중 유클리드 거리를 사용하여 유사도를 판별했으나 이 역시 결과가 좋지 않아 OpenCV 를 쓰지 않고 다른 방안을 고안해보았다.

두번째 접근으로는 Siamese network(이하 삼 네트워크)를 이용해 두 이미지간 유사도를 비교를 하려했다. 삼 네트워크는 클래스의 종류가 많고, 특정 클래스에 대한 사진을 대량으로 구할 수 없을 때 머신러닝을 활용하여 그 클래스를 구분하기 위해 고안된 네트워크이다. 이 네트워크의 단점은 새로운 데이터가 투입 되었을 때 민감함으로 인해 정확한 유사도가 판별되지 않아 다른 방법으로 접근했다.

VGG16 은 Visual Geometry Group(VGG)에서 제안한 Convolutional Neural Network(CNN) 모델이다. 이 모델은 깊은 층의 Convolutional Layer 와 Max Pooling Layer 로 구성되어 있어 이미지 특징을 추출하는데 매우 효과적이다.

우리는 각 그림 class 마다 유사도 기준이 다르기에, 비슷한 것과 그렇지 않은 것을 500 ~ 1500 장 사이로 뽑아보고 유사도 기준을 잡았다.



## ②-1. 모델 주요 내용

그림판 파일을 플라스크가 파일 타입 할 수 있도록 웹에서 그린 그림판 파일을 Flask 가 그림 file type 을 알 수 있도록 하는 것에 중점을 두고 코드를 구현했다.

```
@app.route("/vgg")
def similarity_image():
    q, p_path, h_path, sim = random_sim()
    return render_template('1st_test.html', q=q, p_path=p_path, h_path=h_path, sim=sim)

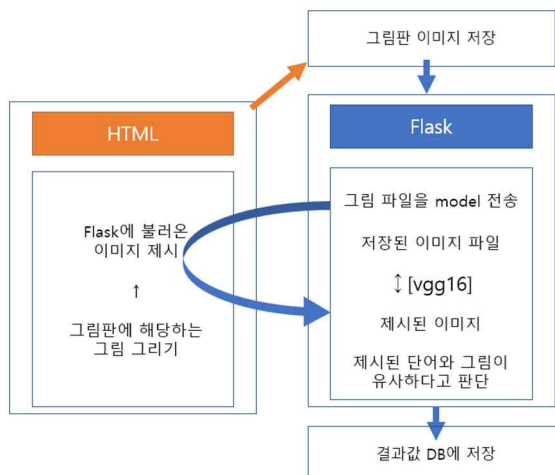
# @app.route("/sim_test", methods=["POST"])
# def sim_test():
#     p_path = str(request.form['p_path'])
#     sim = float(request.form['sim'])
#     return render_template('sim_test.html', p_path=p_path, sim=sim)

@app.route("/image-similarity", methods=["POST"])
def image_similarity():
    f = request.files['file']
    img_path = 'static/1/img/img.png'
    f.save(img_path)
    p_path = str(request.form['p_path'])
    sim = float(request.form['sim'])

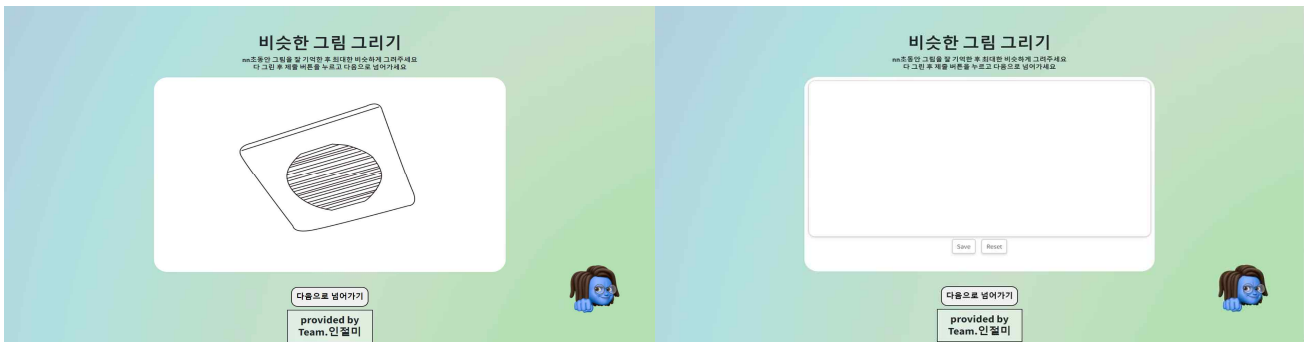
    features1 = get_image_feature(p_path)
    features2 = get_image_feature(img_path)

    cosine_similarity = np.dot(features1, features2) / (np.linalg.norm(features1) * np.linalg.norm(features2))
    print(cosine_similarity)
    if cosine_similarity >= sim:
        return render_template('success.html')
    else:
        return render_template('fail.html')
```

## ● 최종 모델 :



### ③ web 에 부착 시 모습



## 2) 글자 색 맞추기

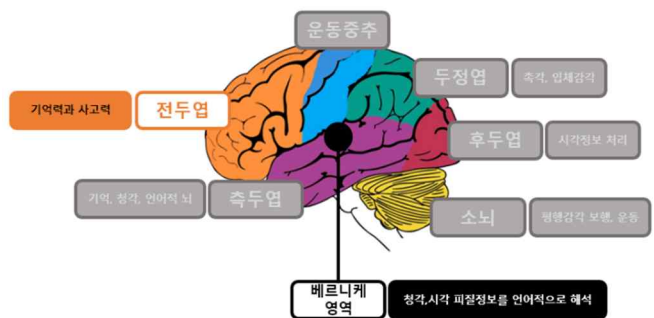
: 과제에 대한 반응시간이 검사자의 주의에 따라 달라지는 효과 또는 이러한 현상을 이용하는 검사

### (1) 개발 취지

스트루프 검사란, 심리학에서 스트루프 효과(Stroop effect) 또는 스트루프 검사(Stroop test)는 과제에 대한 반응시간이 검사자의 주의에 따라 달라지는 효과 또는 이러한 현상을 이용하는 검사를 말한다. "파랑", "초록", "빨강"과 같은 글자와 이 글자가 나타내는 의미인 실제 색상이 일치하지 않을 경우, 즉 "빨강"이 빨간색으로 프린트되어 있지 않고 노란색으로 프린트되어 있을 때 글자의 색을 말하는 데 더 오랜 시간이 걸리며 잘못 말하는 경향이 생기는 현상이다. 이 효과는 1929 년 처음 독일에서 보고 되었으나, 1935 년 이를 영어로 최초 보고한 John Ridley Stroop 의 이름을 따 명명되었다. (출처 : 위키백과)

### (2) 해당 검사의 기대 효과 :

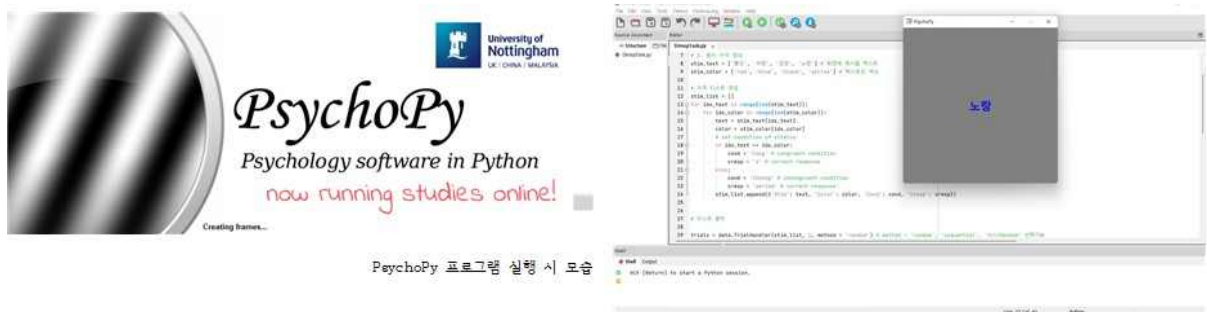
- ① 단순, 복합 반응시간
- ② 억제 / 탈억제
- ③ 전두엽 및 베르니케 영역 자극
- ④ 집행 기술 - 처리 속도 확인



### (3) 개발

1. 개발 환경 : VScode 1.74.3, PsychoPy 2022.2.5, Python 3.10
2. 개발 작업 :

## ① PsychoPy를 통한 프로그램 구현



- 코드는 for 문과 if 문을 사용하여 반복되는 상황에서 자동으로 window 창에 실행되도록 함
- 코드를 완성하고 프로그램 실행을 완료하였으나, 웹(flask)에 부착이 안 된다는 사실을 확인하고, VScode 에서 실행이 가능한 순수 Python 만으로 만들어 보기로 함.

## ② 응용 프로그램을 사용하지 않고 Python 만으로 프로그램을 구현

```
def on_submit():
    global prev_time, index
    input_text = input_box.get().strip()

    writer.writerow([
        time.time() - prev_time,
        "1/%03d.png" % (index + 1),
        correct_answers[index],
        input_text,
        '정답' if correct_answers[index] == input_text else '오답'
    ])

    input_box.delete(0, tk.END)
    prev_time = time.time()
    index += 1

    if index >= len(correct_answers):
        root.destroy()
        return

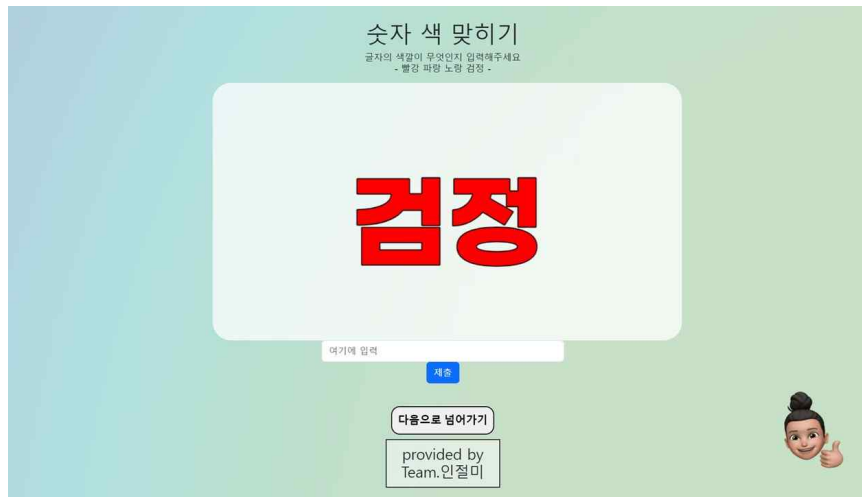
    new_image = Image.open("1/%03d.png" % (index + 1))
    new_photo = ImageTk.PhotoImage(new_image)
    label.configure(image=new_photo)
    label.image = new_photo
```

### ② -1. 코드 주요 내용

- Python tkinter : tkinter 는 GUI 에 대한 표준 Python 인터페이스이며 Window 창을 생성할 수 있다. (import tkinter as tk)
- Pillow(PIL) library : 이미지 분석 및 처리를 쉽게 할 수 있는 라이브러리, 다양한 이미지 파일 형식을 지원하며, 강력한 이미지 처리와 그래픽 기능을 제공하는 이미지 프로세싱 라이브러리의 한 종류이다. (\* 함수 선언으로 불필요하게 중복되는 if 문을 간소화함.)



## ② web 에 부착 시



### 3) 제시 단어 그리기

: 정보가 시각적일수록 우리는 더 쉽게 인지하고, 더 오래 기억, 그림 우월성 효과를 이용한 사례

#### (1) 개발취지

2018 년 캐나다 워털루 대학 연구 결과, 노년기 기억력 증진에 글쓰기 보다 그림 그리기가 더 효과적이라고 한다. 해당 연구 결과, 젊은 층과 노년층 모두에서 그림을 그려서 외운 사람이 더 많은 단어를 기억하는 것으로 나타났다. 특히 기억력 증진의 효과는 노년층에서 두드러졌다.

연구에 따르면 그림 그리기는 시각적, 공간적, 언어적인 요소는 물론, 운동적 요소까지 포함된 공감각적인 표현 방식이어서 다른 기억법보다 우월하다. 여기서 그림을 얼마나 잘 그리는지는 중요하지 않다. 일반적으로 새로운 정보를 기억하는 능력은 나이를 먹으면 쇠퇴한다. 해마, 전두엽 등 기억을 관장하는 뇌의 중요한 부위의 기능이 떨어지기 때문이다. 반면, 이미지와 그림 등을 표현하는 데 관련된 뇌의 시각 및 공간 처리 영역은 노화 또는 치매에 큰 영향을 받지 않는다. 멜리사 메이드 연구원은 “그리기를 통한 기억법은 아직 손상되지 않은 뇌 부위를 활용하기 때문에 기억력 장애와 함께 언어 능력이 급속하게 쇠퇴하는 치매 환자에게도 도움이 된다.”고 말했다.

우리는 위의 연구결과를 바탕으로 생활 속에서 많이 접할 수 있는 객체 14 가지를 선정하여, 해당 검사자가 제시어를 통해 그림을 그리는 활동을 통해 경도 인지 수진자를 넘어 생활 속에서 인지능력 향상을 꾀하기 위해 해당 프로그램을 기획하게 되었다.

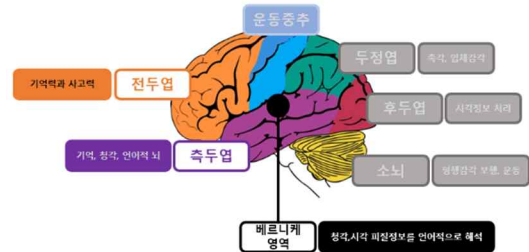
(출처 : Drawing as an Encoding Tool: Memorial Benefits in Younger and Older Adults - Experimental Aging Research )

#### (2) 해당 검사의 기대효과

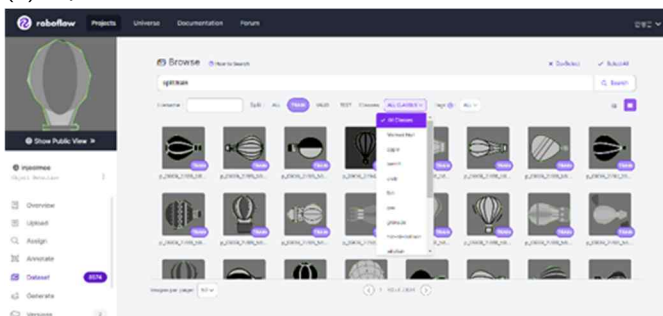


- 프로그램의 목적 : 외부에서 주어지는 정보를 선택적으로 주의를 집중하여 장기인지 밖으로 정보를 불러일으킬 때 얼마나 정확하게 도출해 내는가

- ① 운동속도
- ② 소근육 조절
- ③ 집행 기능
- ④ 빠른 의사 결정



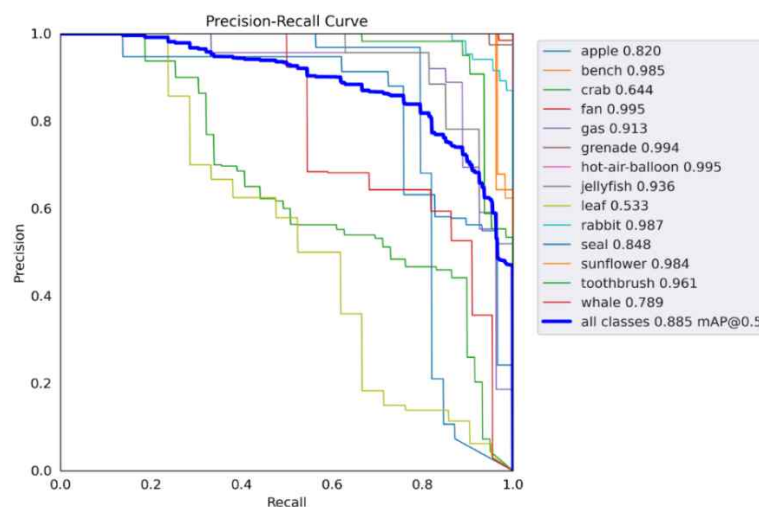
### (3) 개발



(약 10000 개의 image dataset 을 이용했다.)

#### ① roboflow 를 통한 이미지 데이터 전처리, 라벨링, 확인 및 모델 훈련을 자동화

- a. 이미지 데이터 업로드 : Roboflow 에서 이미지 데이터 셋을 업로드
- b. 라벨링 : Roboflow 에서 이미지에서 객체를 식별하고 각 객체에 라벨을 할당
- c. 데이터 전처리 : 이미지 데이터 셋을 전처리하여 모델 훈련에 사용할 수 있는 형식으로 변환
- d. 모델 훈련 : 모델을 훈련하고 테스트
- e. 모델 사용 : 훈련된 모델을 사용하여 새로운 이미지에서 객체를 식별
- f. 모든 훈련을 마친 class 중 정확도가 높은 객체 (8/14) 선정

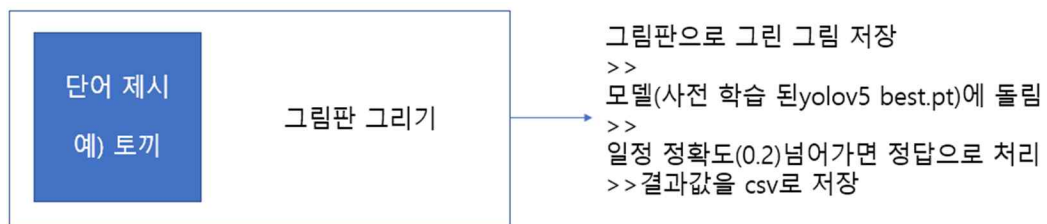


② PyTorch HUB를 이용하여 app.py 함수에 들고 올 수 있도록 구현하고, 사전에 돌린 모델에서 best.pt를 들고와 이미지 데이터셋을 들고 올 때 일일이 학습할 필요 없이 학습이 되도록 함

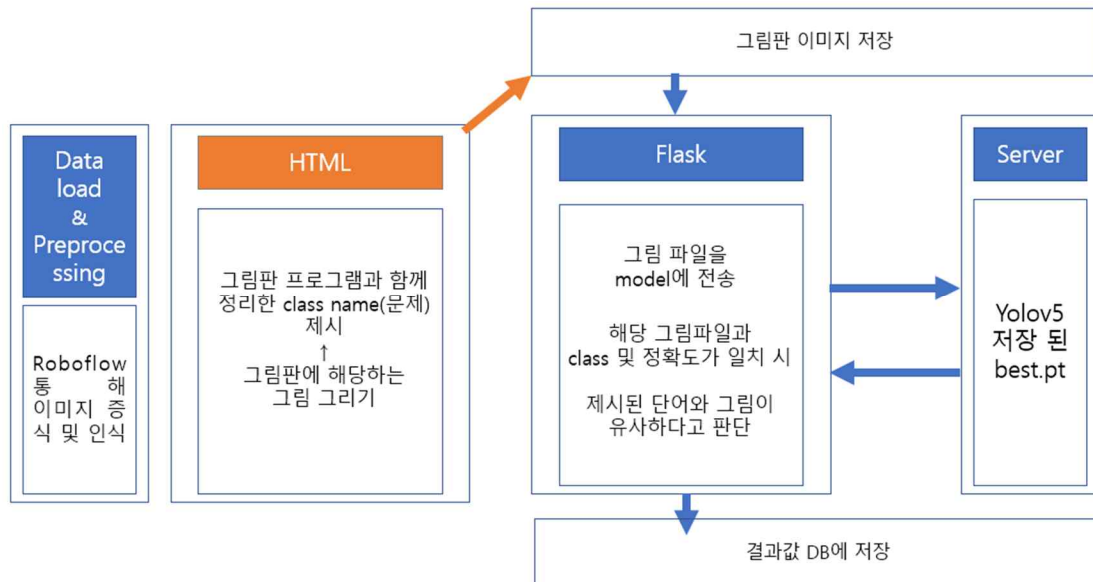
```
@app.route("/", methods=["GET", "POST"])
def predict():
    # Model(YOLOv5 종속 항목 설치)
    model = torch.hub.load('ultralytics/yolov5', 'custom', path = 'best.pt', force_reload =True)
    # Image
    img = ['C:\\Users\\admin\\Desktop\\최종 프로젝트\\글,그림\\벤치.png']

    # 추론
    results = model(img)
```

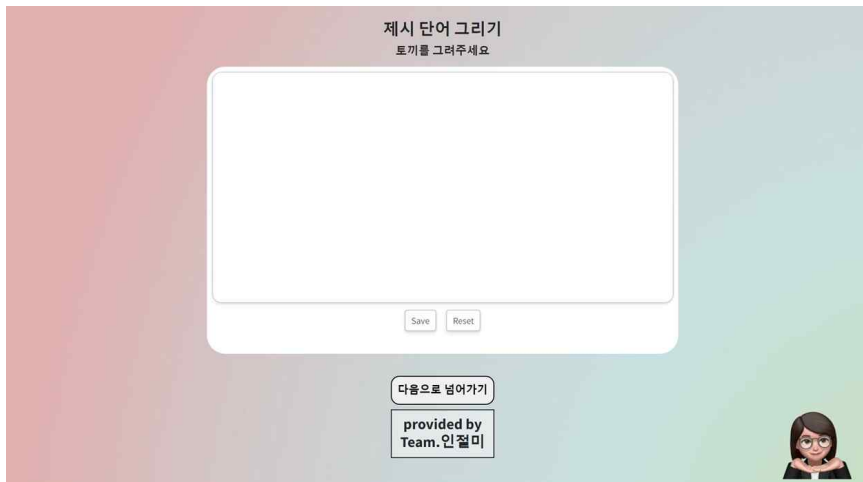
- 초기 구상도 :



- 최종 모델 :



### ③ web 에 부착 시



### 4) 다른 그림 찾기

: 서로 다르면서도 비슷한 이미지 간의 차이점을 알아내는 퍼즐의 일종

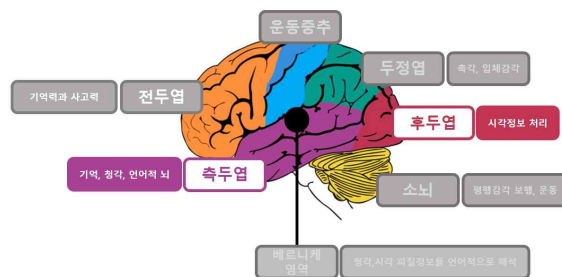
#### (1) 개발 취지

다른 그림 찾기란, 서로 다르면서도 비슷한 이미지 간의 차이점을 알아내는 퍼즐의 일종이다. 둘 이상의 것을 눈으로 비교해 보면서 나란히 배치하거나, 두 물체를 겹쳐서 보거나, 또는 이미지를 번갈아 보면서 수행할 수 있다. 화면 내에서 제시하는 단어의 문맥을 파악하고 난 뒤, 보기에서 제시하는 여러 이미지 가운데 특정하게 다른 이미지를 찾아내는 것을 기준으로 잡게 되었다.

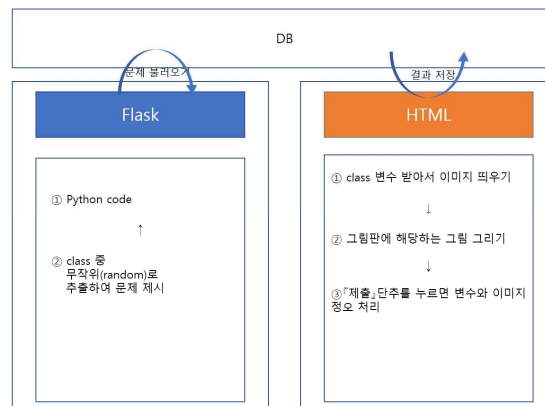
#### (2) 해당 검사의 기대효과

- 프로그램의 목적 :

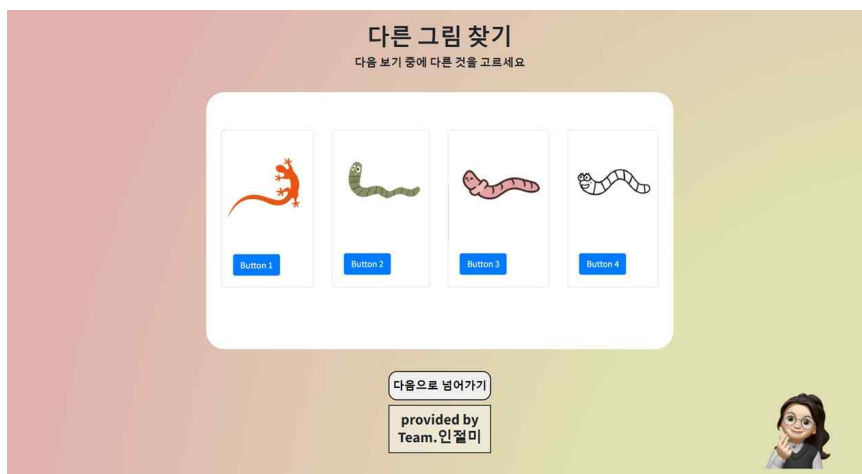
- ① 시각적 학습
- ② 선택 반응 시간
- ③ 시각적 및 지연된 상기



### (3) 개발



### ③ web 에 부착 시



### 5) 숫자 순서 맞추기

: 실생활에서 종이 타일을 이용해 사용 가능, 게임을 할 때는 타일의 위치를 기억(주어진 시간 내에서 순서대로 숫자 기억하기)

#### (1) 개발취지



[출처 : 'Chimp Memory Test' on YouTube/ [Chimp Ayumu touchinal](#)]

화면에서 띄워주는 숫자열을 첫 단계에서 바로 따라 외우고 그 숫자를 순차대로 누르도록 시행 되게 한다. 이 프로그램은 시각적 단기 기억 및 주의력과 주로 관련 있으며 문화적 영향을 거의 받지 않는 특징을 가지고 있어 인지 능력을 환경에 영향을 받지 않고 검사할 수 있다는 장점이 있다.

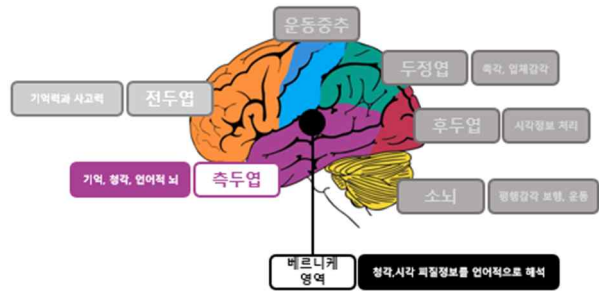
순차 연결을 한다는 것은 작업기억의 주 요소 검사로 계열화, 정신적 조작, 주의력, 시공간적 형상화, 처리속도와 관련이 있다.

우리는 이를 쉽게 접근하기 위해 위의 영상과 같이 언제 어디서든 쉽게 접근할 수 있는 게임을 만들고자 했다.

## (2) 해당 검사의 기대효과

- 프로그램의 목적 : 연속 수행 능력 및 시각적 기억

- ① 지속되는 주의 집중
- ② 선택 반응 시간
- ③ 즉각적인 및 지연된 상기
- ④ 충동성 억제



## (3) 개발

### 1. 개발 작업

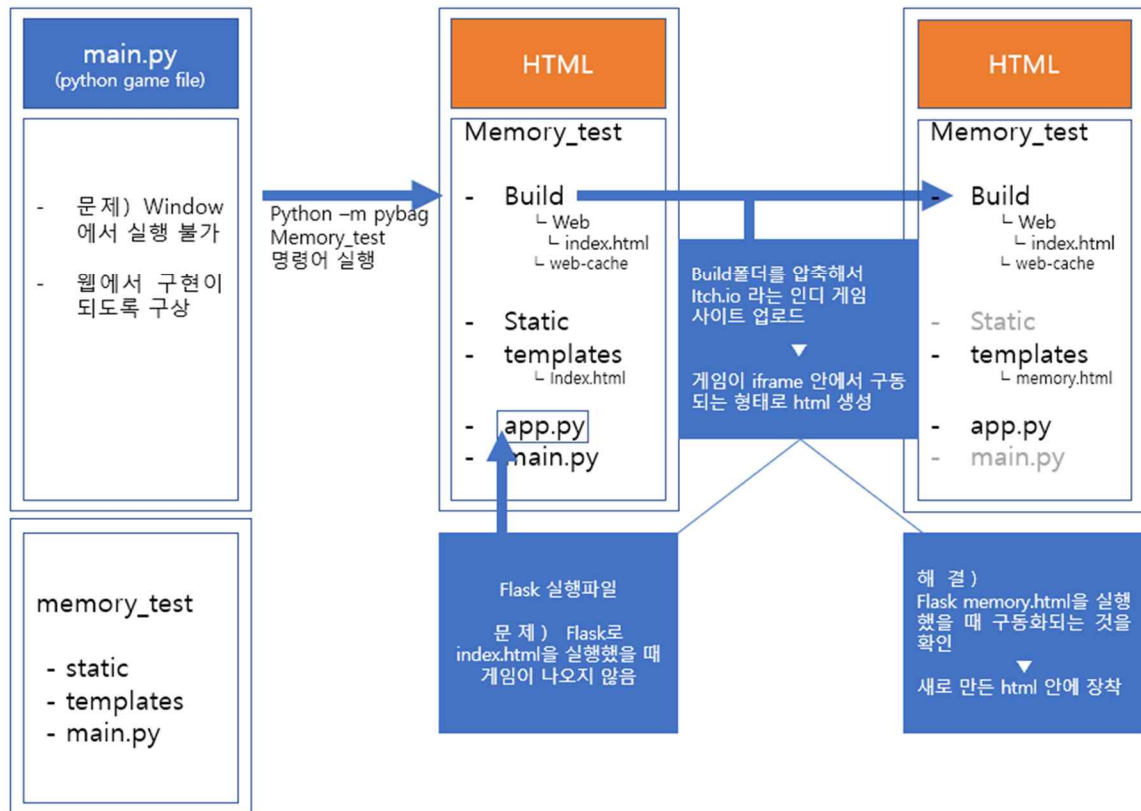
- ① library 중 pygame 은 비디오 게임 작성용으로 설계된 크로스 플랫폼 python 모듈 세트이다.
- 해당 코드는 youtube 나도코딩에서 나오는 강의를 기반으로 작성했다. 그대로 코드를 하기 보단, 레벨 제
- 초기 구상도

해당 기본 프로그램에서, ㉠레벨 10 까지 제한 ㉢ 해당 결과를 csv 로 저장하여 web 에 부착시키는 것을 목표로 삼았었다. 하지만, 해당 library(pygame)를 이용하여서는 web 부착하는 것이 불가능 하여 다른 방안을 찾아보기로 했다.

### ② EasyOCR

EasyOCR 은 문자 영역 인식(Detection)과 문자 인식(Recognition) 기능을 모두 수행하는 프레임워크이다. 숫자 순서 맞히기의 결과창을 스크린샷하고 EasyOCR 을 사용하여 스크린샷 이미지에서 결과값에 해당하는 텍스트를 추출한다.

• 최종 모델 구현도



② web 에 부착 시



## 6) 문장 따라 말하기

: 들은 문장을 언어적으로 처리하고 기억하여 말하는 능력이 필요, 문법적으로 단순한 문장부터 차츰 길고 구문이 복잡한 문장으로 구성

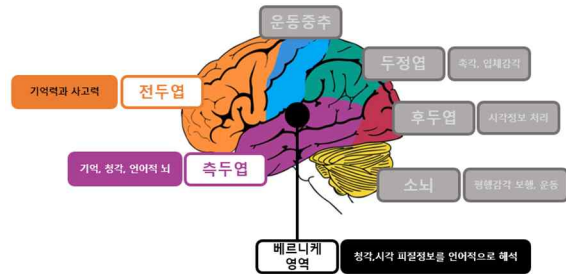
### (1) 개발 취지

작업 기억이란 정보의 유지 및 처리와 조작에 기여하는 인지기제를 뜻한다. 이것은 어휘습득, 구어 산출, 읽기 발달, 언어 이해 등 전반적인 언어 처리에 영향을 미친다. 인지저하를 겪고 있거나 작업기억의 손상으로 인해 언어적 처리, 저장 및 산출 등에 있어서 어려움을 겪는다. 특히 개인의 작업기억 용량 안에서 청각 지각 처리부터, 음운 분석, 출력 맵핑(mapping) 및 산출의 다양한 인지 능력을 필요로 하는 따라 말하기의 경우 인지저하를 겪는다면 낮은 수행력을 보일 수밖에 없다.

이를 판별하기 위하여 K - MoCA(한국판 몬트리올 인지평가) 검사지 중에 '문장 따라 말하기'를 차용하여 프로그램을 개발하였다. 채점 기준은 ETRI 에서 제공하는 API 를 이용하여 제시 문장과 검사자가 수행한 결과가 얼마나 유사한지를 보고 최대한 검사자가 정확하게 말하는 것으로 삼았다.

### (2) 해당 검사의 기대 효과 : 언어적 기억, 연속 수행 능력검사

- ① 언어 학습
- ② 단어 재인
- ③ 즉각적인 및 지연된 상기
- ④ 충동성 억제



### (3) 개발

#### ① 개발 작업

- 모델 주요 내용 :

녹음 파일 생성 후 음성을 인식해 텍스트로 변환하기 위해(Speech To Text) ETRI 에서 제공하는 API 를 이용하였다. 결과값을 추출할 때 원하는 부분만 추출하기 위해서 변수를 만든 다음, list indexing 을 했다.

```
http = urllib3.PoolManager()
response = http.request(
    "POST",
    openApiURL,
    headers={"Content-Type": "application/json; charset=UTF-8", "Authorization": "accessKey"},
    body=json.dumps(requestJson)
)

print("[responseCode] " + str(response.status))
print("[responBody]")
print(str(response.data, "utf-8"))

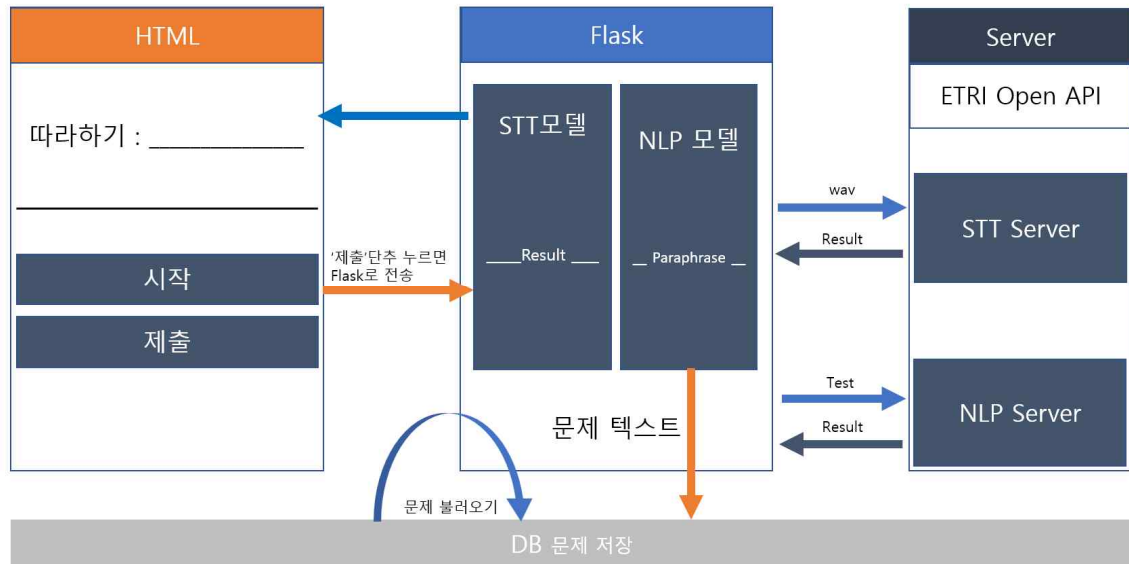
NLP_String = str(response.data, "utf-8")
NLP_List = NLP_String.split('')
print(NLP_List)

NLP_reuslt = NLP_List[-2]
# NLP_reuslt = NLP_target[:-1]
print(NLP_reuslt)
```

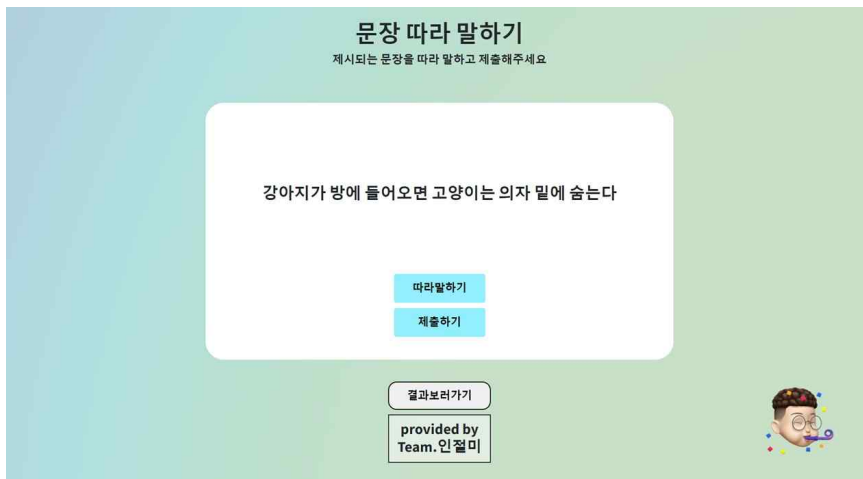
유사도 NLP API 사용 이유 : 사용하는 사람 사투리 or 발음이 어눌할 수 있으니 유사도 모델을 도입해서 이러한 문제점을 해결하고자 했다.

```
# 유사도 검사 NLP Open API
#-----
openApiURL = "http://aiopen.etri.re.kr:8000/ParaphraseQA"
accessKey = "f0f9fd15-daef-4655-b516-d7a9711c696a"
sentence1 = String_sound
sentence2 = String_target
```

● 모델 구현도:



② web 에 부착 시





### 3. 프로그램 시연

#### 1) 웹 개발

##### 1 웹 공통 사항

###### 1.1 UI/UX

- 웹을 전체적으로 훑어볼 수 있도록 기타 기능은 오른쪽 상단에 위치
- 빠른 테스트를 원하는 사람을 위해 시작하기 버튼을 중심 왼편에 위치
- 테스트 및 화면에 집중할 수 있도록 테스트 페이지부터는 기능들을 중심에 위치
- 모든 페이지에 대표 로고 각인



###### 1.2 색상 및 캐릭터

- 신뢰와 함께 편안한 마음으로 테스트에 임할 수 있는 컬러 선정.
- 무겁고 딱딱한 분위기를 지향하고 좀 더 친근함을 강조하기 위해 캐릭터 삽입

##### 2 Home.html



###### 2.1 색상

- 신뢰감을 줄 수 있는 짙은 푸른색 & 흰색 사용.
- 필독 내용 이외에는 옅은 폰트컬러 활용.

###### 2.2 전체 분위기

- 간단히 우리 테스트를 볼 수 있는 샘플 이미지 첨부
- 친근감과 편안함을 위해 home.html에도 이미지 적극 활용

### 3 About(us/test).html

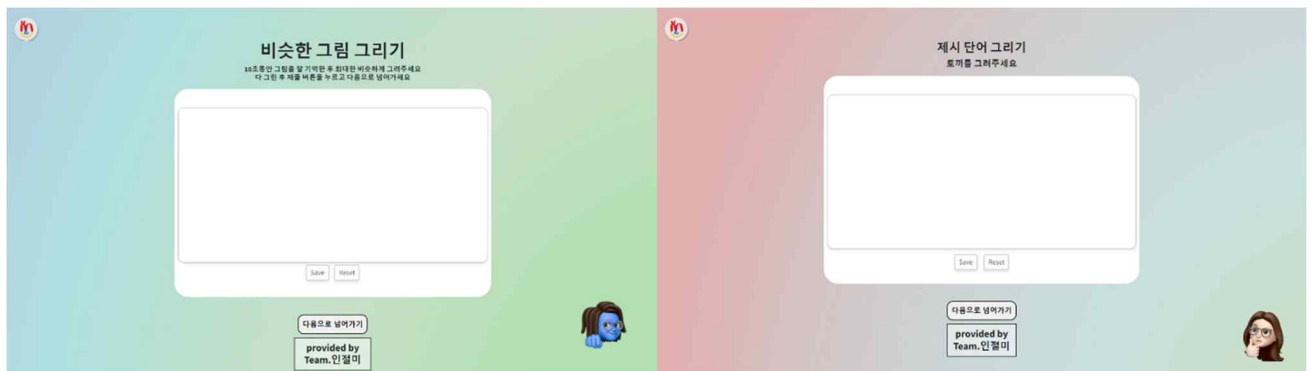


#### 3.1 전체 분위기

- 단정하지만, 우리의 사이트와 팀을 알 수 있도록 상세히 기술.
- 한 눈에 보기 편하게 이미지 적극 활용
- 웹의 취지에 맞게 경도인지저하에 대한 자료 기술
- 팀의 아이덴티티 표출

### 4 Test 페이지

#### 4.1 Canvas.js



- 1, 3번 html에 적용. 그림을 그릴 수 있는 js코드 구현
- 정확한 진단을 위해 전체 지우기로 그림 지우기 가능

#### 4.2 chatbot기능

- 약간의 '이스터에그'처럼 테스트에 있어 재미와 작은 힌트를 주는 영역.
- 개발자들의 캐릭터를 넣어 좀 더 친근함을 유도

## 5 Dashboard.html



### 5.1 UI/UX

- 전체적으로 눈이 편안한 녹색을 메인으로 진행
- 중요한 검사결과를 가장 큰 사이즈로 중심에 배치, 중요도에 따라 위치와 순위 설정

### 5.2 human-centered project에 근간한 음악 제공

- 구글 음악을 통한 저작권 문제없는 음악 선정, 페이지 진입 시 자동재생
- 사용자가 자유롭게 오디오 컨트롤 가능

### 5.3 DB연동 그래프

- 사용자의 점수 파악 및 기존 이용자의 평균치 확인

### 5.4 경도인지장애 정보 및 체크리스트

### 5.5 경도인지저하와 관련된 링크 및 자료 다운로드

- 전문자료 다운로드 및 신빙성 있는 뉴스/백과사전 하이퍼링크 기능

## 2) 서버 개발

### - 추진 서버

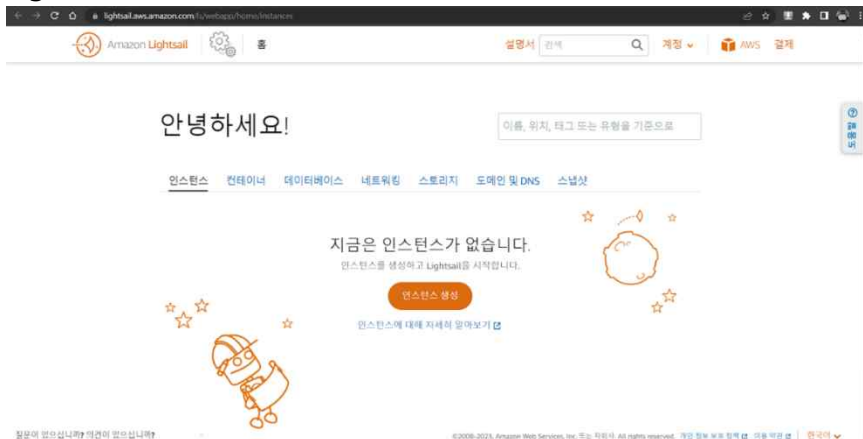
1. AWS S3 : JavaScript 에서 생성된 파일을 AWS S3 에 업로드하고 Flask 에서 다운로드를 받아서 모델에 추가한다.
2. AWS LightSail : 프로젝트를 통체로 AWS 에 담아서 LightSail 에서 프로젝트를 실행한다. (문제 : 용량이 작아 실행이 안된다)
3. MobaXterm 도구를 사용해 현업 멘토님의 회사에서 제공해주신 서버에서 프로젝트와 SSL 를 탑재하고 프로그램을 실행되게 한다.

Back-end : Flask 를 이용해, 해당 Game, Model 과 Front-End 간에 연동과 구현

Front-End : Back-end 과 연동되게 구현하고 간단한 UI 를 만들고 기능들이 정상적으로 돌아가는 것을 확인하고 Web 팀에 넘긴다

#### ① 사용 서버 :

인지능력 테스트를 우리만 가능하게 하면 의미가 없기에 서버를 사용하여 전세계에 공개해보기로 했다. 서버는 AWS(Amazon Web Service) 중 웹 호스팅을 간단하게 할 수 있는 Lightsail 을 사용하기로 했다.



상단의 이미지는 로그인 한 후의 AWS Lightsail 의 홈페이지이다. 저 사이트에서 인스턴스를 생성하면 우리들만의 서버를 만들 수 있게 된다. 인스턴스는 Ubuntu 플랫폼으로 구축했고 인스턴스를 생성한 후에는 AWS 에서 제공해주는 SSH 를 활용하여 서버 환경을 구축했다.

우선 Python 이 설치되어있는지 확인했다.

```
ubuntu@ip-172-26-0-67:~$ python3
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

파이썬이 설치되어 있고 버전이 3.8.2 인걸 확인했다. 그 후 하기와 같이 경로들을 생성했다.

```
ubuntu@ip-172-26-0-67~$ mkdir project
ubuntu@ip-172-26-0-67~$ mkdir venv
ubuntu@ip-172-26-0-67~$ ls
project  venv
```

project 에는 우리의 프로젝트가 venv 에는 가상환경을 구축할 생각이다. 가상환경을 구축하기 전 그냥 만들려고 하면 오류가 생기기에 우선 기본 환경을 업데이트 해준다.

```
ubuntu@ip-172-26-0-67~$ sudo apt-get update
```

● 설치한 패키지 설명 :

패키지 관리 시스템의 리포지토리에서 사용 가능한 패키지와 종속성을 포함한 해당 버전의 목록을 업데이트

이렇게 하면 패키지를 설치하거나 업그레이드할 때 최신 버전을 사용할 수 있다.

```
ubuntu@ip-172-26-0-67~$ sudo apt-get install python3-virtualenv
```

● 설치한 패키지 설명 :

Python3 가상환경 구축에 필요한 패키지

이걸 설치하지 않으면 추후에 설치할 pip3 install python3-venv 가 설치되지 않는다.

```
ubuntu@ip-172-26-0-67~$ sudo apt-get install build-essential libssl-dev libffi-dev python-dev
```

● 설치한 패키지 설명 :

- ✓ build-essential - 데비안 패키지를 빌드하는 데 필요한 필수 도구 세트가 들어 있는 패키지
- ✓ libssl-dev - OpenSSL 라이브러리 개발 파일, SSL 암호화에 필요
- ✓ libffi-dev - 파이썬용 C 확장을 만드는 데 사용되는 외부 함수 인터페이스 라이브러리 개발 파일
- ✓ python-dev - Python 확장을 구축하기 위한 헤더 파일, 라이브러리 및 개발 도구

설치하지 않으면 추후 pip3 install Flask-Migrate 설치에 오류가 생긴다.

이제 기본 환경 구축이 끝났으니 가상 환경을 구축하면 된다.

```
ubuntu@ip-172-26-0-67~$ sudo apt install python3-venv
```

● 설치한 패키지 설명:

- Python3 가상환경을 구축하는 패키지

cd 를 통해 가상환경을 구축할 venv 로 경로를 변경한다.

```
ubuntu@ip-172-26-0-67:~$ cd venv
ubuntu@ip-172-26-0-67:~/venv$
```

경로변경을 확인한 후 python3 -m venv ijm 를 입력한다.

```
ubuntu@ip-172-26-0-67:~/venv$ python3 -m venv ijm
ubuntu@ip-172-26-0-67:~/venv$
```

오류 없이 커맨드 줄이 다시 뜨면 성공이며, 다시 한 번 더 경로를 변경해준다.

```
ubuntu@ip-172-26-0-67:~/venv$ cd ijm/bin
ubuntu@ip-172-26-0-67:~/venv/ijm/bin$
```

경로 변경을 확인한 후, activate 로 가상환경을 실행한다.

```
ubuntu@ip-172-26-0-67:~/venv/ijm/bin$ . activate
(ijm) ubuntu@ip-172-26-0-67:~/venv/ijm/bin$
```

가상환경이 정상적으로 실행되면 windows 의 cmd 처럼 환경명이 앞쪽에 뜬다.

다음은 가장 중요한 패키지 설치이다. 패키지들의 경우 'requirements.txt'에 설치할 패키지들을 정리한 후 이 파일을 사용하여 설치할 것이다. 하지만 따로 설치 해야 하는 패키지가 있다. 그게 바로 'Pytorch'이다. 우리 모델에는 'Yolov5'가 있기에 필요한 패키지이다.

패키지들 중 'Pytorch'를 먼저 설치한다. 'Pytorch'의 경우 "<https://pytorch.org/get-started/locally/>" 에서 각 플랫폼 별로 명시해둔 방법에 따라 설치하지 않으면 오류가 생기기에 'requirements.txt'에서 제외해뒀다. 하단의 코드가 Linux 환경에서 Stable 버전의 Python 언어와 cpu 를 사용하는 'Pytorch'를 설치하는 코드이다.

설치를 완료한 후 "<https://github.com/Liarhouse/requirements.git>"에 저장해둔 'requirements.txt'를 git clone 을 활용하여 다운받은 후 설치할 것이다.

```
(ijm) ubuntu@ip-172-26-0-67:~/venv/ijm/bin$ git clone Wn
https://github.com/Liarhouse/requirements.git
Cloning into 'requirements'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.04 KiB | 354.00 KiB/s, done.
```

다운받은 후 ls 가 문제없이 받아졌는지 확인해본다.

```
(ijm) ubuntu@ip-172-26-0-67:~/venv/ijm/bin$ ls
Activate.ps1  activate  activate.csh  activate.fish  easy_install  easy_install-3.8  pip  pip3
pip3.8  python  python3  requirements
```

위와 같이 'requirements'라는 폴더가 생성된걸 볼 수 있다. 이 안에 'requirements.txt'가 있기에 하단과 같이 커맨드창에 입력해주면 패키지 설치가 시작된다.

```
(ijm) ubuntu@ip-172-26-0-67:~/venv/ijm/bin$ pip install -r Wn requirements/requirements.txt
```

이제 환경 설정은 끝났다. 다음은 우리의 프로젝트를 다운받으면 된다. 맨 처음에 만들어둔 'project' 폴더로 돌아가야 된다.

```
(ijm) ubuntu@ip-172-26-0-67:~/venv/ijm/bin$ cd ~/project
(ijm) ubuntu@ip-172-26-0-67:~/project$
```

경로 변경을 확인한 후 git clone 을 사용하여 프로젝트를 다운받는다.

```
(ijm) ubuntu@ip-172-26-0-67:~/project$ git clone
https://github.com/tenderisthenightt/Final_Project_IJM.git
```

다운이 완료되면 이제 FLASK\_APP 환경 변수를 설정해야 한다.

```
(ijm) ubuntu@ip-172-26-0-67:~/project$ export FLASK_APP=app
```

그 후엔 경로를 프로젝트 폴더로 잡아준 후 누구나 들어올 수 있도록 host 를 0.0.0.0 으로 세팅한 후 서버를 오픈하면 된다.

```
(ijm) ubuntu@ip-172-26-0-67:~/projectFinal_Project_Ijm$ flask run --host=0.0.0.0
```

이렇게 하면 세상에 우리들의 프로젝트가 공개된 것이다. 우리의 서버는 도메인은 없다. 도메인을 사용하게 되면 유료이기 때문이고 그래서 그냥 IP 를 잡아서 사용해야 한다.

이제 "13.209.25.171:5000"를 입력하면 이 세상의 누구든 해당 서버에 접속할 수 있을 것이다.

#### 4. 프로젝트 소감

##### 1) 프로그램 확장성

##### (1) 비즈니스

대한치매협회에 따르면, 아직도 경도인지장애에 대한 일반인들의 인식은 여전히 부족한 것으로 조사됐다. '경도인지장애에 대한 대국민 인식조사' 결과에 따르면 전체 응답자 중 58%는 '경도인지장애라는 용어를 들어본 적 없다. 오늘 처음 들어본다'라고 답했다. '경도인지장애가 치매를 예방할 수 있는 중요시기인지를 전혀 알지 못한다'는 응답자는 73%에 달했다.

이러한 인지 저하는 우울증·조현증 등과 같이 질병분류상 F 코드로 묶어 경증질환으로 치부되어 분류되고 있다. 또한 중증화 가능성을 염두에 두고 있어 보다 과학적인 분류 체계가 필요하다고 대한치매학회에서 알리고 있다.

인간의 나이는 시간이 지날수록 기억력의 감퇴가 발생한다. 노화가 진행되면서 나타나는 자연스러운 현상은 '인지 저하'라는 울타리에서 더 나아가 치매로 갈 수 있는 길로 이어지기도 한다. 우리는 이러한 모습을 일상생활에서 벗어나지 않도록 돕기 위한 인지 프로그램을 만들고 싶었다. 노화로 인해 일어나는 증상만 확인을 하는 것이 아니라 초로기 치매, 경도인지장애, 기억력 저하 주의 등 나이와 성별을 가리지 않고 발생하고 있는 인지 저하에 관한 모든 증상들을 다루고 싶었기 때문이다. 어려운 일이 생기면 자연스럽게 보호자를 찾는 모습, 나이가 지날수록 가족에게 짐이 된다는 무거움을 덜 느끼게끔 하고자 이러한 프로그램을 제작하게 되었다.

서로가 서로에게 큰 힘이 되어줄 수 있는 MCI 프로그램을 사용해 인지 저하가 무거운 주제가 아니라는 것을 알려주기 위해 권장한다.

## (2) 확장성

모든 프로그램은 사용자의 니즈를 파악하고 적합한 전략을 통해 활동을 마무리하는 전반의 과정을 이루고 있다. 이러한 프로그램은 대부분의 산업에서 요구되는 활동으로 이루어져 있으며, 이에 우리는 기업, 소비자 그리고 국가까지 뻗어갈 수 있는 총 세 가지의 확장성을 기대하고 있다.

### ① B2B(Business to Business)

그 중 가장 먼저 B2B(Business to Business)인 기업 간의 거래를 설명하고자 한다. 기업을 대상으로 하는 구매자와 판매자 모두 기업 단위이기 때문에 거래의 주체가 되는 서비스의 단위 또한 큰 편이다. 이는 기업에서 사용하는 병원을 예시로 생각을 하면 이해하기 쉽다. 병원 내에 존재하는 상담을 위해 병원 측에서는 따로 검사를 진행하고 난 뒤 일정 시간 분석을 통해 검사결과를 환자에게 알리는 등의 번거로운 절차를 진행하고 있다. 이러한 기관에서의 특성에 맞춰 프로그램을 사용하게 된다면, 플랫폼의 사용만으로 별다른 과정을 거치지 않고서도 이용을 할 수 있을 것이라 예상된다.

### ② B2C(Business to Customer)

다음은 B2C(Business to Customer)로 소비자와의 거래를 설명한다. 서비스를 직접 사용하는 소비자를 대상으로 하기 때문에 기존의 소비보다 깊이를 가늠할 수 없을만큼 깊고, 보이지않을만큼 넓은 크기를 가지고 있다. 이는 더 나아가 무궁무진한 시장성을 기대할 수 있다는 표현을 나타내기도 한다. 이러한 소비자의 특성을 통해 프로그램을 사용하고자 한다면, 병원에 가지 않아도 편리한 검사 방식을 통해 보다 적절한 프로그램을 개인이 사용할 수 있을 것이라 예상하고 있다.

### ③ B2G(Business to Government)

마지막으로 B2G(Business to Government)인 정부와의 거래를 이야기한다. 사회는 인지 저하에 대한 장벽을 크게 느끼고 있다. 이러한 사회를 '친화 사회'로 구축하기 위해서는 전문인력을 양성하고, 민간에서 관리 체계를 구축하는 것을 필요로 하게 되는데 이때에 관련 산업을 육성할 필요성을 정부에서 느끼게 될 것이다. 이를 통해 전문적인 진료를 필요로 하기 전 가능성 여부를 판단할 수 있는 프로그램이 존재하게 된다면, 보다 더 나은 사회적 비용과 부담이 줄어듦 것으로 예상된다.



또 다른 접근으로는 공익성을 추구하는 방향이다. 현재 치매 전 단계 '경도인지장애' 254 만명 시대에 도달하면서 조기 대응하면 사회적 비용 감소를 기대해볼 수도 있다 이야기하고 있다. 하지만, 사회는 끊임없이 많은 것을 요구하고, 늘 새로운 것을 추구한다. 그럼에도 불구하고 혼자만의 길을 걷는 것이 아닌 함께하는 미래를 그려가는 것을 목표로 걷고 있다. 그러기 위해 우리는 미래를 함께 하기 위해 공익성을 추구하는 방향으로 나아가는 것이 최종적인 모두의 목표로 세울 수 있을 것이다.

## 2) 프로그램 후기 및 소감

1. 임지영 : 처음으로 한 달 반이라는 기간과 8 명의 사람들과 작업을 하는 거라 초반에는 조금 혼란스러웠지만 점점 결과물이 보여서 뿌듯하고 즐거웠습니다. 끝까지 잘 마무리 했으면 좋겠습니다. 화이팅!!
2. 김동률 : 꽤나 긴 시간동안 팀원들과 함께 골머리 앓아가면서 풀리지 않던 문제를 풀려고 애를 쓰고 그리고 결국 해결하면서 얻었던 쾌감을 잊을 수 없을것 같다. 그리고 다양한 모델뿐만 아니라 프론트 관련한 정보들을 많이 알수있어서 얻어가는 것이 많았던 프로젝트였다. 부팀장으로써의 역할을 더 충실히 하지 못한부분에 대한 아쉬움이 남는다.. 다양한 감정의 소용돌이가 있었지만 그럼에도 보람있었던 프로젝트였다고 생각한다.
3. 박혜빈 : 이번 프로젝트는 길다면 길고 짧다면 짧은 시간이었다. 그 시간동안의 나는 원하는 결과물에 대한 희열도 느껴보고, 보고싶지 않았던 과정을 통해 쓴 맛도 맛보게 되었다. 혼자 울적해하기도 하고, 이리저리 방향을 하기도 하면서 어떻게 해야하나 하는 고민을 하면서 길을 찾아다닌 것 같다. 하지만 그럴 때마다 매번 카페 나들이와 산책을 함께 해준 팀원들에게 무한한 감사를 표현하면서 프로젝트를 마치고 싶다. 정말 수고 많았다!
4. 박희현 : 주어진 일을 수행하고 더욱 발전하는 일은 어렵더라도 보람차지만, 했던 것을 무마하고 새롭게 다시 하는 것은 손을 놓기가 너무 힘들었다. 하지만 더 나은 발전을 위해 해야 함을 이해하고 내가 했던 것을 아까워하지 않는 것을 배울 수 있는 귀중한 시간이 되었다.
5. 백지은 : 6 개월이 번개같이 지나갔네요. 선득하게 심장이 뛰는 순간도 있었고, 1 초가 영원처럼 느껴지는 순간도 있었어요. 그리고 벌써 마칠 시간이네요. 새로운 것들을 많이 배우고 좋은 사람들도 만나서 행복했습니다. 이 경험들을 발판삼아 더 나은 사람이 되도록 노력하겠습니다. 모두 감사했습니다!
6. 송승우 : 재미있고 좋은 경험이었습니다. 완성도를 위해 프로젝트 기간이 좀 더 길었으면 좋겠습니다. 해당 과정을 통해서 많이 배우고 성장할 수 있어서 좋았습니다. 이번 프로젝트가 너의 디딤돌이 되기를 바랍니다.
7. 안정은 : 여럿이 함께 의견을 도출하고 서로의 마음을 구현한다는 것이 뜻깊고 유익한 시간이었습니다. 커뮤니케이션의 중요함을 알 수 있는 소중한 기회였습니다.
8. 정동환 : 안녕하세요. 재밌었습니다. 감사합니다.

## 2) 참고 문헌 / 사이트

### ① 유사도 검사

- (1) AdamW(2017), Decoupled Weight Decay Regularization
- (2) VGG16 논문 : Very Deep Convolutional Networks for Large-Scale Image Recognition
- (3) Image similarity estimation using a Siamese Network with a contrastive loss,  
[https://keras.io/examples/vision/siamese\\_contrastive/](https://keras.io/examples/vision/siamese_contrastive/)
- (4) Siamese Neural Networks (삼 네트워크) 개념 이해하기,  
<https://tyami.github.io/deep%20learning/Siamese-neural-networks/>

### ② 스트루프 게임

- (1) 신경과학 그리고 심리학 실험을 위한 Python & PsychoPy programming(저자 : 남종호)

### ③ 그림그리기

- (1) PyTorch: YOLOv5 모델을 이용한 이미지 객체 탐지, <https://foss4g.tistory.com/1646>
- (2) Load YOLOv5 from PyTorch Hub ★, <https://github.com/ultralytics/yolov5/issues/36>

### ④ 틀린 그림 찾기

- (1) 틀린그림찾기로 알아보는 치매 위험성?- 서울브레인 신경과 이일근 원장,  
<https://youtu.be/NybadAlMBwU>

### ⑤ 기억력 게임

- (1) itch.io 에 올린 memory\_test, <https://jabullae.itch.io/memory-test>
- (2) pygame 을 웹으로 구동시킬수 있는 pygame github, <https://github.com/pygame-web/pygbag>

### ⑥ 문장 따라 말하기

- (1) HTML 오디오 태그 (Audio, iframe) 음악 자동재생, <https://hongku.tistory.com/364>
- (2) Audio+Video+Screen Recording using RecordRTC, <https://www.webrtc-experiment.com/RecordRTC/>
- (3) RecordRTC is WebRTC, <https://github.com/muaz-khan/RecordRTC>

### ⑦ Web

- (1) Bootstrap, <https://getbootstrap.kr/>
- (2) W3Schools Online Web Tutorials, <https://www.w3schools.com/>
- (3) gentelella, <https://github.com/ColorlibHQ/gentelella>
- (4) [WEB] HTTP 상태코드를 알아보자,  
<https://ssungkang.tistory.com/entry/WEB-HTTP-상태코드를-알아보자>
- (5) 코딩꼬적꼬적 - 파이썬(Python)/Flask,

<https://scribblinganything.tistory.com/category/%ED%8C%8C%EC%9D%B4%EC%8D%AC%28Python%29/Flask>

⑧ 기타

(1) 경도인지장애, 치매 전조증상일수도,

<https://mdtoday.co.kr/news/view/1065567329425911>

(2) “치매 전 단계 ‘경도인지장애’ 254 만명 시대...조기 대응하면 사회적 비용 줄인다”,

<https://biz.chosun.com/science->

[chosun/bio/2022/09/19/G7A5PJCL6JHSNPNFVDDKSSGK24/?utm\\_source=naver&utm\\_medium=original&utm\\_campaign=biz](https://biz.chosun.com/science-chosun/bio/2022/09/19/G7A5PJCL6JHSNPNFVDDKSSGK24/?utm_source=naver&utm_medium=original&utm_campaign=biz)

(3) B2B 영업과 B2C 영업 이해하기,

<https://www.robertwalters.co.kr/career-advice/b2b-vs-b2c-what-is-the-difference.html>