

대한상공회의소 광주인력개발원

머신러닝 및 딥러닝 활용 프로젝트 (얼굴 인식)

TEAM 3조 팀명 : 박가네

박성빈, 박의용

목 차



프로토타입 모델



01. 프로젝트 계획

02. 구현

03. 요구사항 정의 및 분석

04. 설계

05. 후기

01. 프로젝트 계획

구분	내용
1. 주제선정 배경	개인 PC 해킹 등 보안이 취약해짐에 따라, 인공지능을 이용하여 웹 브라우저를 제어하는 시스템 구현 방식을 논의 하였다.
2. 개요	머신러닝, 딥러닝으로 얼굴인식 학습 모델 구축한다. 얼굴인식으로 PC 브라우저 창을 제어한다. (구현)
3. 구조	서버 인공지능 얼굴 학습 모델 및 데이터베이스 구축 클라이언트 얼굴 인식 시 PC 브라우저 창 제어
4. 기대효과	본인 얼굴 인식으로 PC 브라우저 접근, 보안능력 향상

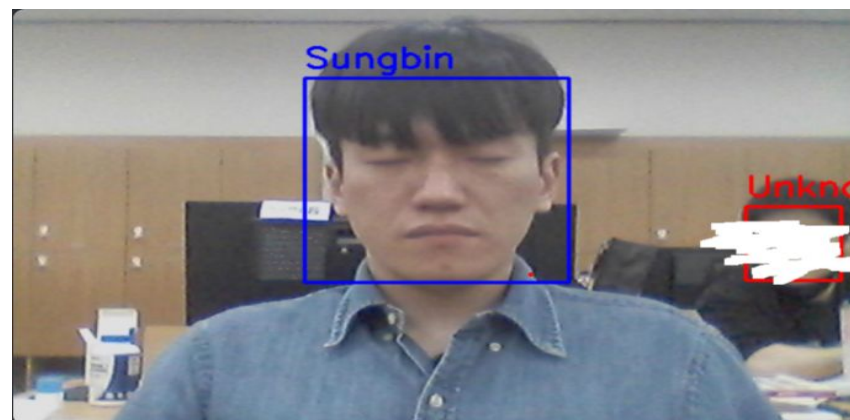
01. 프로젝트 계획

1 - (팀 구성 및 역할)

훈련생	역할	담당 업무
박성빈	팀장	<ul style="list-style-type: none">▶ 서비스 시스템 설계 (주제 선정, 서비스 구조)▶ 서버 시스템 및 클라이언트 시스템 총괄 (DB 구축)▶ 데이터 수집, 머신러닝 및 딥러닝 (얼굴인식)
박의용	팀원	<ul style="list-style-type: none">▶ 서버 , 클라이언트 UI 시스템 구축 (데이터 송, 송신)▶ 데이터 수집, 머신러닝 및 딥러닝 (얼굴인식)

02. 구현

1 - 사람 구별 모델



```
import cv2
import numpy as np
import onnxruntime as ort

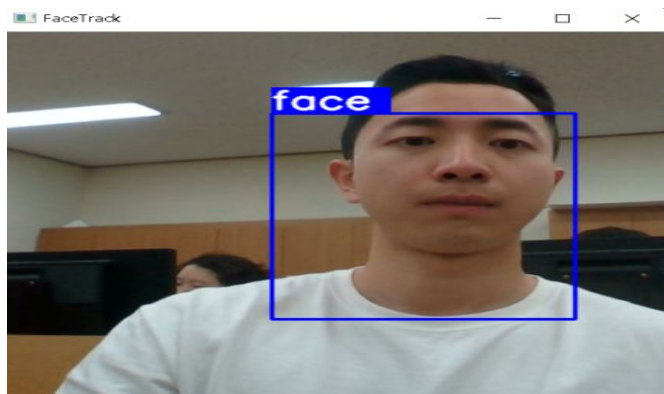
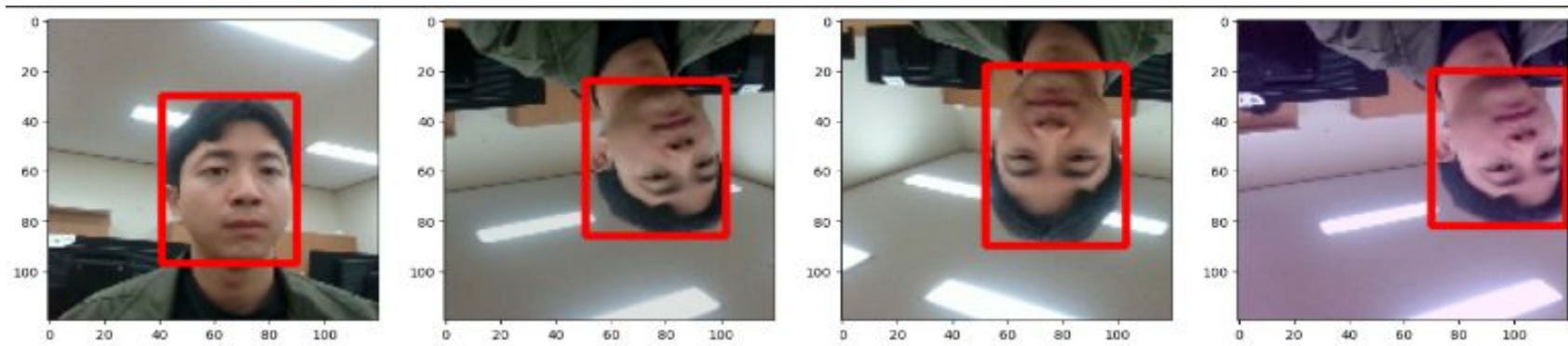
# Load the face detection model
face_detector = cv2.CascadeClassifier('C:/Users/tjdql/Desktop/haarcascade_frontalface_alt.xml')
```

```
# Load the face recognition model
model = ort.InferenceSession('C:/Users/tjdql/Desktop/holy.onnx')
```

얼굴인식은 haarcascade 모델다운로드, 2. 얼굴구별모델은 성빈이와 의용이 사진 3천장으로 구별모델 학습

02. 구현

2 - 얼굴 인식 모델 ([깃허브 링크](#))

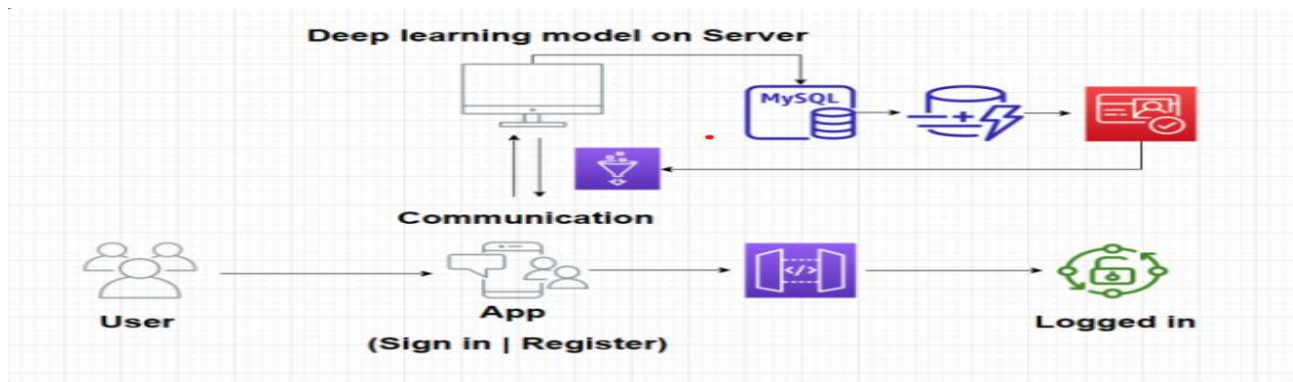


```
import cv2
from tensorflow import keras
import tensorflow as tf
import numpy as np
from tensorflow.keras.models import load_model

facetracker = load_model('uiyongfacetracker.h5')
```

모델 학습할 때 데이터를 얼굴이 아닌 데이터를 포함을 안하고 모두 얼굴 데이터만 학습시켜서 박스가 안사라지는 문제점 발견.

03. 요구사항 정의 및 분석



얼굴인식을 통한 자동인터넷 제어 프로그램 요구사항 분석서

번호	요약	요구사항	요구분석 내용	정	부	비고
1	러닝	데이터 학습	본인얼굴과 다른사람얼굴로 사진학습 본인얼굴일시 특정신호를 보내고, 본인얼굴이 아닐 시 특정신호를 보내게 학습	박성빈	박의용	
2	서버	서버	로그인 시간 및 로그아웃 시간을 DB에 업데이트	박성빈	박의용	
3	클라이언트	클라이언트 기능	허용된 얼굴이 로그인 시 인터넷 창을 열고, 허용되지 않은 얼굴 시 인터넷 창을 닫음	박성빈	박의용	
4	클라이언트	클라이언트 기능	로그인 시간 및 로그아웃 시간을 서버에 보냄	박성빈	박의용	

04. 설계

(개발 환경 및 사용 라이브러리)

개발 환경	사용 라이브러리
OS	▶ Window 10
Language	▶ Python , C#
Framework	▶ Winform, PyQt5
DBMS	▶ MySQL
Library	▶ Python Channels, TCP/IP, OpenCV, tensorflow
AI	▶ CNN, Keras
IDE(통합개발환경)	▶ Visual Studio, Pycharm, Jupyter Notebook, Google Colab

04. 설계

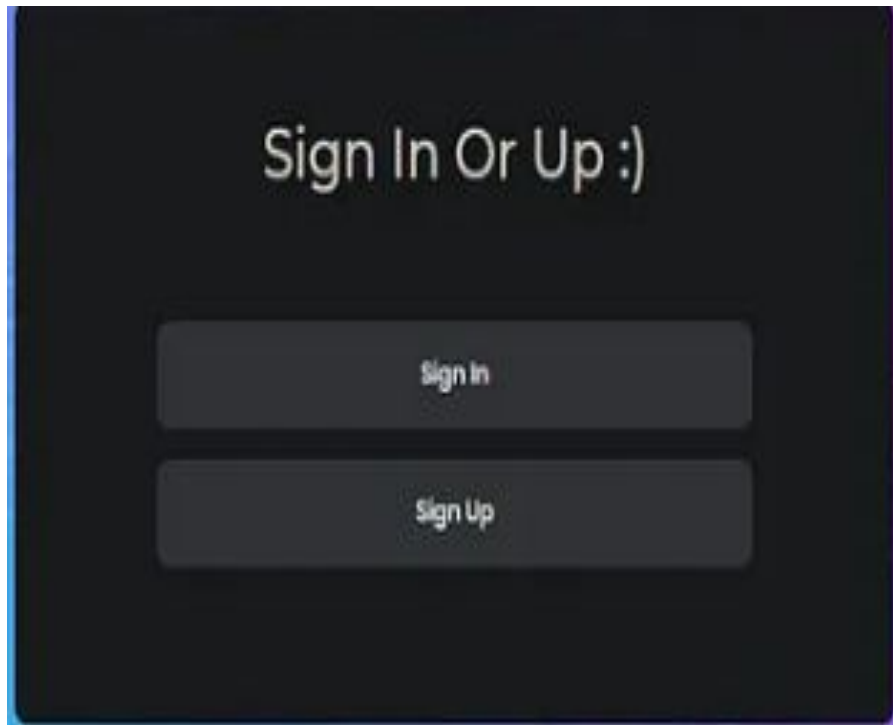
1 - (신경망 구축 및 모델 학습 방법)

단계	내용
1	필요한 라이브러리 설치 및 데이터 가져오기
2	데이터셋 검토 및 이미지 로딩 기능 구축
3	증강되지 않은 데이터 분할
4	이미지 및 레이블에 이미지 확대 적용
5	증강 파이프라인 구축 및 실행
6	라벨 준비, 라벨 및 이미지 샘플 결합
7	기능적 API를 사용하여 딥러닝 구축 (1~7 단계로 신경망 구축)
8	손실 및 옵티마이저 정의
9	모델 훈련하기
10	모델 테스트 및 적용

04. 설계

2- (프로토 타입)

구현기능	<ul style="list-style-type: none"> - 본인의 얼굴 사진 및 여러타인의 얼굴 사진 학습 - 얼굴인식이 허용되는 얼굴 2명을 학습 - 2명은 허용된 얼굴, 나머지는 UNKNOWN으로 인식되게 하기 - 클라이언트에서는 얼굴에 네모박스를 띄우고 이름이 누구인지, UNKNOWN인지 판별케 하기 - 허용된 얼굴이 로그인 버튼을 누르면 해당얼굴 이름, 현재시각을 서버에 보내고 서버에서 들어온 시간을 업데이트, 인터넷 창을 열기 - 허용된 얼굴이 로그아웃 버튼을 누르면 해당얼굴 이름, 현재시각을 서버에 보내고 나간시간을 DB에 업데이트, 인터넷 창 닫기 - 허용되지 않은 얼굴이 로그인 버튼을 누르면 자동으로 인터넷 창이 열려있으면
	<p>닫고, (UNKNOWN, 현재시각)을 서버에 보내고, 서버는 DB업데이트</p> <ul style="list-style-type: none"> - UNKNOWN, 현재시각을 DB에 담는이유는 식별안된사람이 로그인 시도 한것을 인지하기 위함



04. 설계

3- (프로젝트 수행 절차 및 방법)

구분	기간	활동	비고
사전 기획	4/24(월) ~ 5/02(화) (9일)	▶ 프로젝트 기획 및 주제 선정 ▶ 기획안 작성	▶ 아이디어 선정
데이터 수집	5/02(화) ~ 5/04(목) (3일)	▶ 필요 데이터 및 수집 절차 정의 ▶ 데이터 수집	▶ Python
CNN 딥러닝	5/05(금) ~ 5/07(일) (3일)	▶ 데이터 정규화로 특징인 얼굴인식	▶ Python Jupyter Notebook
모델링	5/08(월) ~ 5/10(수) (3일)	▶ 서버, 클라이언트 모형 구현	▶ C#, Winform, Python
서비스 구축	5/11(목) ~ 5/12(금) (2일)	▶ 데스크톱 시스템 설계 및 테스트	▶ 최적화, 오류 수정
총 개발기간	5/02(화) ~ 5/12(토) (11일)	-	▶ 사전 기획 기간 미포함

05. 후기

훈련생	역할	개인 후기
박성빈	팀장	당초 목적은 파이썬에서 ONNX로 얼굴구별모델을 학습시킨 이후에 C#에서 실시간 영상을 캡처해서 어떤얼굴인지 구별하려 했습니다. C#에서 모델을 쉽게 로드하기 위해 ONNX로 만들었는데 C# 관련 지식이 부족하여 모델로드하는데 실패하여 파이썬으로 구현하였습니다. C#에 대한 지식이 부족한거 같아 추가학습 하도록 하겠습니다.
박의용	팀원	딥러닝 모델 구축 이후, 취업 기간과 겹친다는 이유로 제대로 프로젝트 진행을 하지 못했다. 실제 모델을 C# 서버 및 클라이언트에 구축 이후 시스템 까지 구축했어야 했는데 완료하지 못했다. 일정 관리 및 실력 부족으로 여러가지 풀어 내지 못한 점이 많아서 아쉬운 마음이 크다.

감사합니다.