

VIP Detection Sensor

얼굴 인식 영상 딥러닝 포트폴리오



Agenda

01

주제 소개

주제 선정 이유, VIP Detection Sensor 소개, 프로젝트 일정, 자체 평가리스트

02

모델 구현

데이터 수집, 데이터 정제, 모델 훈련, 최적화

03

시뮬레이션

테스트 영상 시청

04

결론

프로젝트 의의, 활용방안, 자체 평가 결과

주제 선정

01 주제설명

흥미도, 구현 가능성, 유용성 3가지 측면을 고려하여 최종적으로 VIP Detection Sensor 라는 주제를 선정하였음. 이는 VIP(유연수 선생님)의 얼굴을 실시간으로 감지하는 것을 목표로 둬.



주제 리스트
VIP Detection Sensor
동물 발자국 식별
토마토 품질 식별
영상에서 브랜드 감지



구분	흥미도	구현가능성	유용성
백광흠	7	10	9
김건태	7	8	7
차호성	10	9	10
은해찬	8	8	8



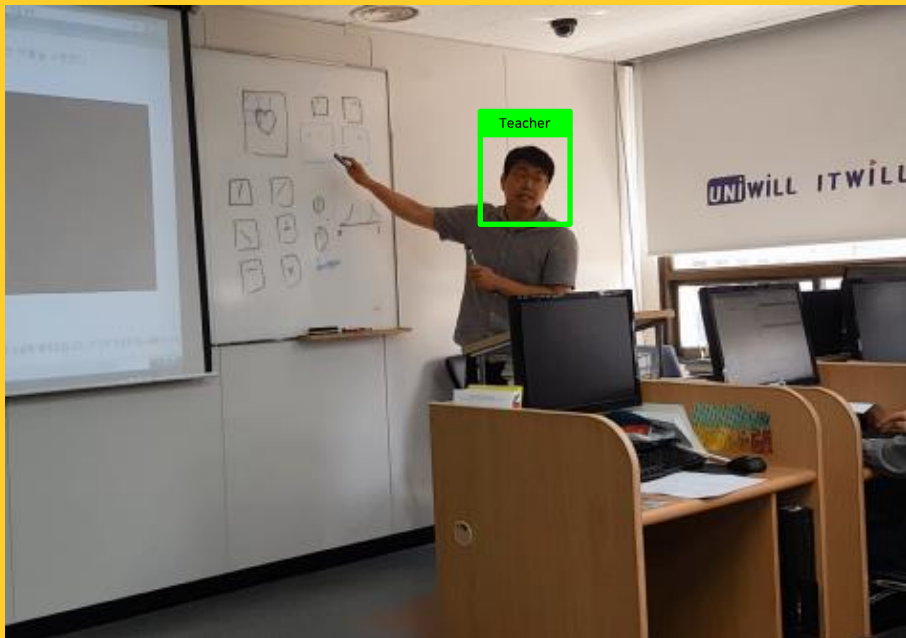
주제 리스트
VIP Detection Sensor
동물 발자국 식별
토마토 품질 식별
영상에서 브랜드 감지



VIP Detection Sensor 소개

01 주제설명

VIP Detection Sensor란, VIP의 얼굴을 실시간으로 웹캠 또는 영상에서 인식하는 모델임.
타겟은 우리반의 유연수 선생님이로 정했으며, 선생님 얼굴을 인식하여 영상에 표시하였음.



VIP Detection Sensor 특징

- 녹화된 동영상 파일이 아닌, **실시간** 타겟 감지 (휴대용 웹캠 이용)
- 인식하고자 하는 **학습을 통해 변경**할 수 있음. 현재 타겟은 유연수 선생님이로 잡았으나, 다른 라벨 학습도 얼마든지 가능함.
- Face-recognition, opencv library를 기반으로 함
: **앞, 옆모습에 상관없이** 타겟을 잘 인식한다는 특징이 있음.
- 모델 학습 속도가 느리고, 동작 시 끊김 현상이 발생
-> **GPU 컴퓨터를 사용**함으로써 속도 문제를 해결



프로젝트 일정

01 주제설명

시간 자원을 적절히 분배하기 위해 프로젝트 일정을 만들어 진행함. 프로젝트 일정 흐름은 크게 모델 제작과 PPT 제작 2가지 방향으로 진행했으며 최적/최악 데드라인을 설정하였음.

데이터 수집

- 예상 소요 기간: 2일
- 최적 데드라인 9/5
- 최악 데드라인 9/6

데이터 정제

- 예상 소요 기간: 1일
- 최적 데드라인 9/6
- 최악 데드라인 9/8

모델 설계 및 구현

- 예상 소요 기간: 3일
- 최적 데드라인 9/11
- 최악 데드라인 9/15

최적화

- 마지막까지 계속 진행

PPT 제작 착수

- 예상 소요 기간: 3일
- D-3

모델 활용방안 구상

- D-2

마무리

- D-1

최종 발표

- 9/20(목)



자체 평가 리스트

01 주제설명

연구소 프로젝트를 완료하기 위한 3가지 평가 목표를 설정함. 각각의 목표들은
① 파이썬 구사능력, ② 모델 적합성, ③ 모델 최적화로 구분됨.

구분		평가 근거	점수
① 파이썬 구사 능력	<ul style="list-style-type: none">파이썬을 활용하여 딥러닝을 위한 데이터 수집을 할 수 있는가?파이썬을 활용하여 모델에 맞는 데이터 정제를 할 수 있는가?	1. _____ 2. _____ . .	__점
② 모델 적합성	<ul style="list-style-type: none">주제에 맞는 적절한 딥러닝 모델을 사용했는가?연구원 모두가 자신만의 모델을 개발했는가?모델의 타당성을 입증할 수 있는가?	1. _____ 2. _____ . .	__점
③ 모델 최적화	<ul style="list-style-type: none">최적모델을 선정하기 위한 과정이 잘 설명되었는가?파라미터 별 모델 평가를 충분히 하였는가?	1. _____ 2. _____ . .	__점





Agenda

01

주제 소개

주제 선정 이유, VIP Detection Sensor 소개, 프로젝트 일정, 자체 평가리스트

02

모델 구현

데이터 수집, 데이터 정제, 모델 훈련, 최적화, 각 연구원 별 모델 4가지

03

시뮬레이션

테스트 영상 시청

04

결론

프로젝트 의의, 활용방안, 자체 평가 결과

모델 제작 과정

02 모델 구현

데이터 수집/정제는 openCV, 구글 웹 스크롤링을 사용하였으며 직접 제작한 코드로 작업을 수행. 모델 훈련, 모델 최적화는 keras python package를 사용하였음.



데이터 수집



데이터 정제



모델 구상 및 훈련



모델 최적화

- OpenCv
- 구글 웹스크롤링

라벨 이미지(선생님)는 동영상 녹화 후 **OpenCv**를 사용했으며, 라벨이 아닌 이미지는 **구글 웹스크롤링**을 통해 수집하였음

- Python으로 전처리 과정 구현

라벨/비라벨 이미지 모두 **64 x 64 사이즈**로 맞춰주는 코드를 제작하여 활용함. 비라벨 이미지의 경우 **비정상 파일**을 제거해 줌.

- Keras

모델 훈련에 keras가 사용됨. 연구원 4명의 각기 다른 **CNN 모델**을 설정하였음. **compile**과 **fit** 함수를 사용하여 비교적 간단하게 모델 훈련 코드를 구현함.

- Keras

Keras 내장 함수인 **evaluate**를 사용하여 모델 정확성을 평가하였음. 다양한 **파라미터**를 **조절**하여 최적의 모델을 찾고자 함.



데이터 수집

02 모델 구현

데이터 수집은 라벨/비라벨로 나누어 진행됨. 라벨 데이터는 선생님의 동영상을 촬영하여 얻었으며, 비라벨 데이터는 구글 웹스크롤링을 통해 얻었음.

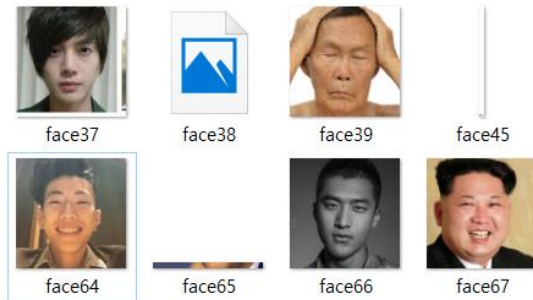
라벨 이미지



라벨 이미지는 **openCV**를 활용하여 영상에서 크로핑을 하여 수집되었음.
그 결과 깨진 파일은 없지만, 픽셀의 크기는 다른 이미지들의 집합이 형성됨. 따라서 **픽셀 크기가 다른 부분에 대해서 데이터 정제 작업이 필요.**



비라벨 이미지



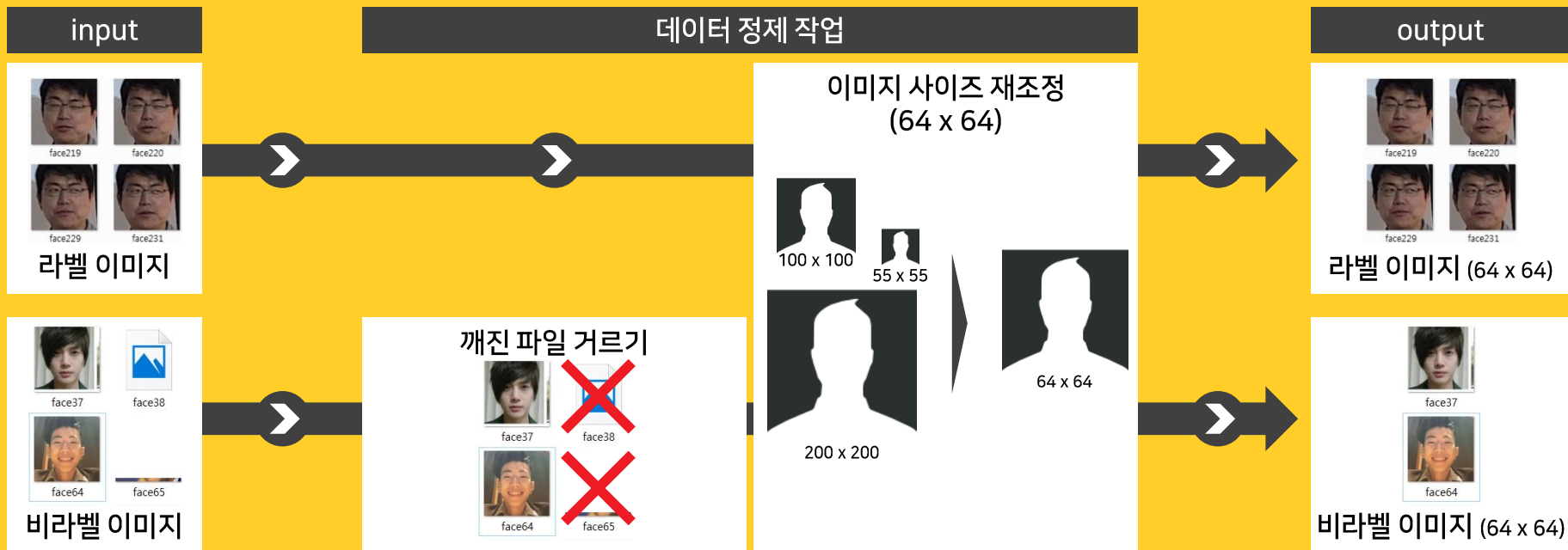
비라벨 이미지는 **구글 웹스크롤링**을 통해 수집되었음.
그 결과 깨진 이미지와 정상인 이미지가 모두 수집되었음.
또한 픽셀의 크기도 모두 다른 이미지 집합이 형성됨.
따라서 **깨진 파일을 걸러내고 픽셀 사이즈를 통일**하는 작업이 필요.

→ 파이썬으로 데이터 전처리 과정을 구현


데이터 정제

02 모델 구현

데이터 정제는 라벨/비라벨 이미지를 구분하여 진행함. 라벨 이미지는 OpenCv의 resize로 크기를 조정했고, 비라벨 이미지는 깨진 파일을 뺀 후 동일하게 이미지 크기를 조정하였음.



4종류의 운영환경을 구축하였음. 각각의 운영환경들은 최신 버전은 아니지만, 수많은 시행착오와 트러블 슈팅 끝에, 가장 호환이 잘 되는 것으로 확인되었음.

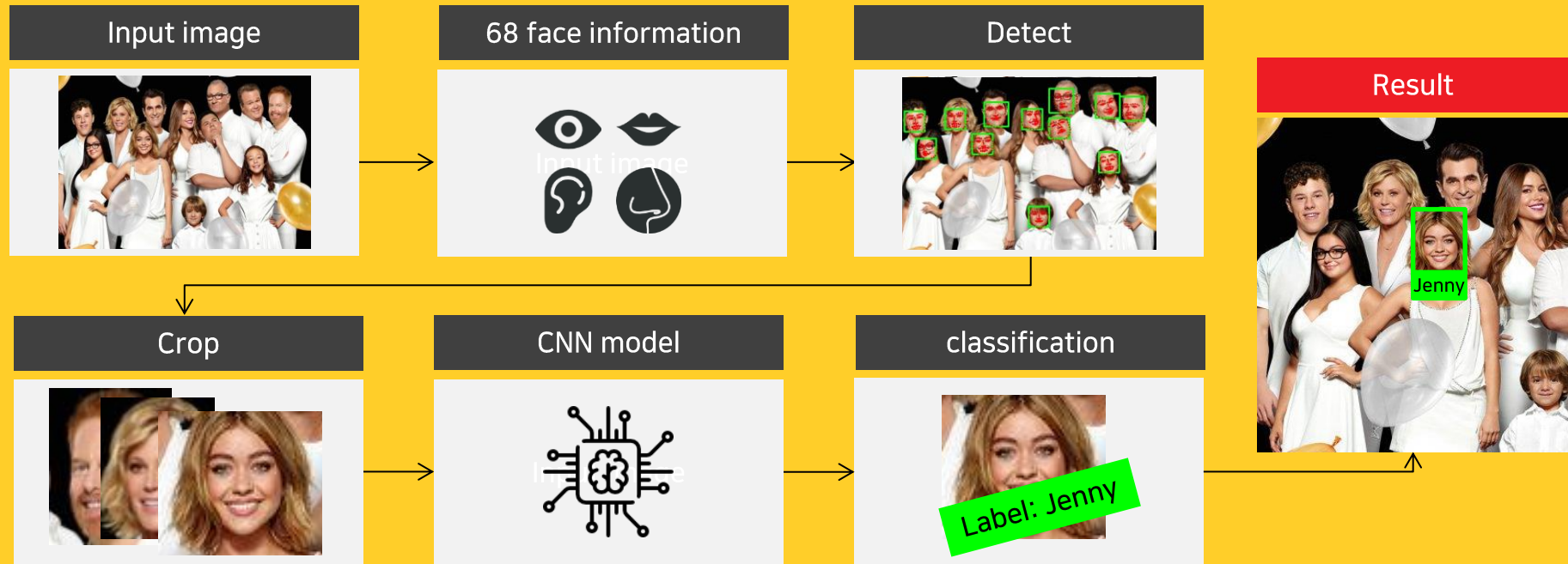
운영체제	GPU 환경	언어 환경	플랫폼 / 소프트웨어
 Ubuntu 16.04	 NVIDIA CUDA toolkit9.0	 Python 3.5.6	 Anaconda Python 3.5 ver.
	 NVIDIA CUDA toolkit9.0		 OpenCV 3.43.18
	 cuDNN		 19.15.99



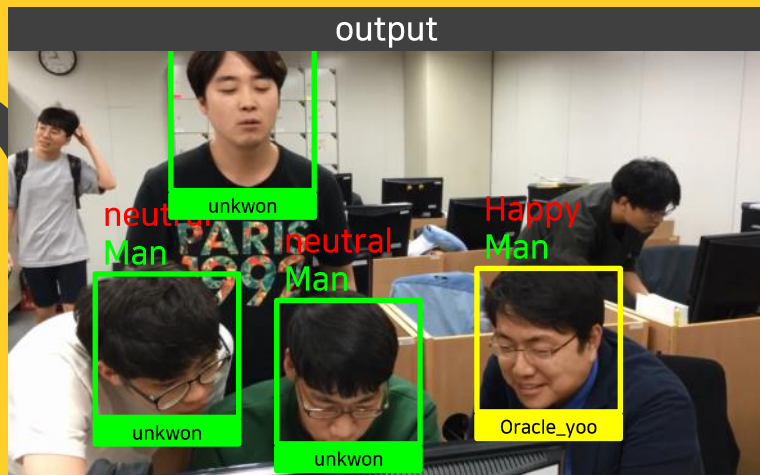
동작원리

02 모델 구현

Face_recognition을 이용한 얼굴 식별 → 식별한 얼굴을 크로핑 → 학습시킨 cnn모델을 이용해 라벨 값 반환 → 식별한 얼굴에 대한 감정/성별 분석 → 화면에 출력



[얼굴인식 → 타겟(선생님)인지 아닌지 판별 → 감정/성별 판별] 프로세스가 진행됨.



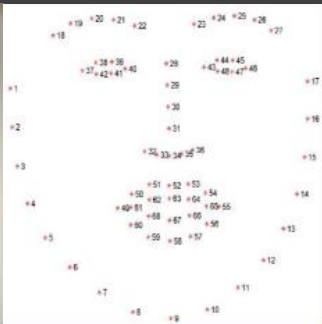
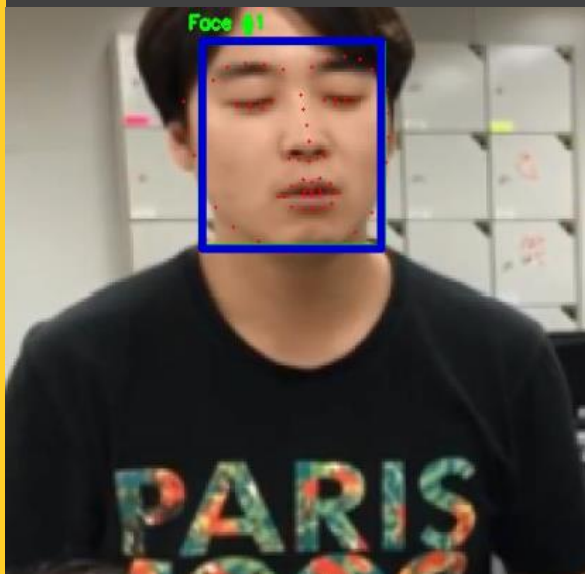
모델 흐름도_1.얼굴영역인식

02 모델 구현

얼굴 영역은 Face_recognition 이라는 라이브러리를 통해 잡아내었음.

Face_recognition은 얼굴의 눈썹, 눈, 코, 입, 턱을 랜드마크화 하여 얼굴을 인식하는 원리.

자료 사진



얼굴엔 총 68개의 랜드마크가 존재하는데,
이를 통해 얼굴을 식별 및 라벨을 판별 할 수도있다.

해당 파이썬 코드

```
boxes = face_recognition.face_locations(rgb,  
                                         model=args["detection_method"])
```

#이미지에서 얼굴을 인식하여 좌표를 리스트 형으로 저장

```
result, oy = model.predict(image)
```

학습된 모델의 predict 함수를 이용해 라벨값, 확률 반환

** 설명을 위한 과정이 생략된 코드입니다.



모델 흐름도_2.라벨 판별

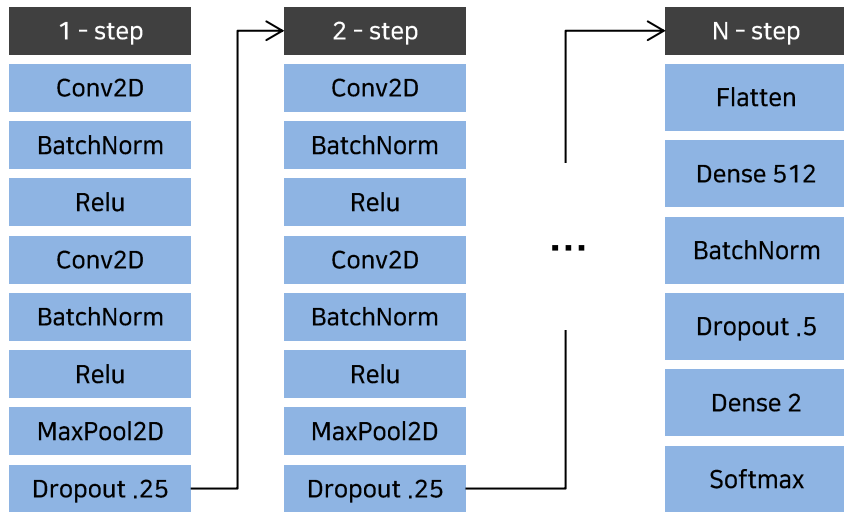
02 모델 구현

라벨 판별은 자체적으로 학습한 모델을 통해 잡아냄. 이 부분에서 연구원들이 개발한 CNN이 적용되었으며, 각자의 모델에서 쓸만한 파라미터를 추출해 최종 모델을 도출하였음.

자료 사진



모델 흐름도



모델 구상 및 훈련 요약

02 모델 구현

모델은 CNN을 사용했으며, 파이썬의 keras를 이용하여 훈련시켰음. 각각의 방법을 사용하여 총 4가지 모델을 만들었음. 이후 모델을 훈련하여 정확도를 도출함.

- 사용한 모델: rmsprop
- 활성화 함수: relu
- 배치 정규화: o
- 가중치 초기값 설정: he
- 정확도: 0.91

- 사용한 모델: SGD
+ Momentum
- 활성화 함수: relu
- 배치 정규화: o
- 가중치 초기값 설정: he
- 손실률/정확도: 0.88

- 사용한 모델: Adagrade
- 활성화 함수: relu
- 배치 정규화: o
- 가중치 초기값 설정: he
- 손실률/정확도: 0.94

- 사용한 모델: Adam
- 활성화 함수: relu
- 배치 정규화: o
- 가중치 초기값 설정: he
- 손실률/정확도: 0.89

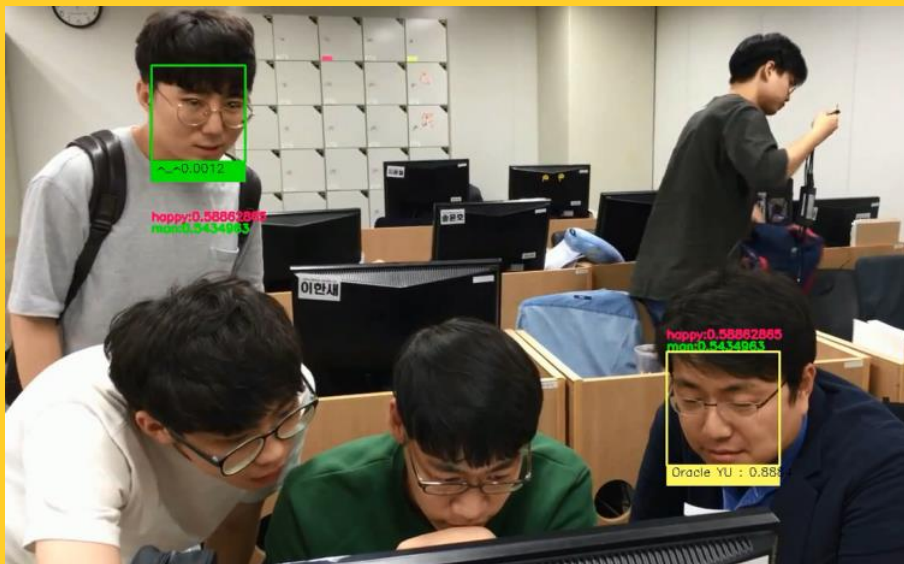


모델 흐름도_3.성별/감정 판별

02 모델 구현

성별/감정 판별은 외부에서 만들어진 성별, 감정 모델을 각각 가져와서 적용하였음.
이를 통해 얼굴로 인식한 모든 오브젝트에 대하여 성별, 감정이 표시되도록 함.

자료 사진



해당 파이썬 코드

```
emotion_label_arg =  
    np.argmax(emotion_classifier.predict(gray_face))  
emotion_max =  
    np.max(emotion_classifier.predict(gray_face))  
emotion_text = emotion_labels[emotion_label_arg]  
  
gender_prediction = gender_classifier.predict(rgb_face)  
gender_max =  
    np.max(gender_classifier.predict(rgb_face))  
gender_label_arg = np.argmax(gender_prediction)  
gender_text = gender_labels[gender_label_arg]
```

감정과 성별의 라벨값을 반환하는 코드입니다.





Agenda

01

주제 소개

주제 선정 이유, VIP Detection Sensor 소개, 프로젝트 일정, 자체 평가리스트

02

모델 구현

데이터 수집, 데이터 정제, 모델 훈련, 최적화, 각 연구원 별 모델 4가지

03

시뮬레이션

테스트 영상 시청

04

결론

프로젝트 의의, 활용방안, 자체 평가 결과

시뮬레이션 동영상 시청

03 시뮬레이션

타겟(유연수 선생님)을 노란색 영역으로 잘 표시하는지 시뮬레이션 했음. 또한 라벨 판별 이외에도 사람들의 감정과 성별이 프레임 위에 표시되도록 했음. 해당 영상은 유튜브에 업로드 하였음.



<https://youtu.be/e52dZfQ9-AA>



보너스 영상

03 시뮬레이션

타겟(유연수 선생님)과 유사인물인 전용준(E-스포츠 캐스터)을 대상으로, 라벨 판별 검사를 실시함.
그 결과, 전용준 영상에서 일부를 선생님의 인물로 인식했지만, 대체적으로는 구분을 잘 하였음.



[유연수 선생님]

VS



[전용준 캐스터]





Agenda

01

주제 소개

주제 선정 이유, VIP Detection Sensor 소개, 프로젝트 일정, 자체 평가리스트

02

모델 구현

데이터 수집, 데이터 정제, 모델 훈련, 최적화, 각 연구원 별 모델 4가지

03

시뮬레이션

테스트 영상 시청

04

결론

프로젝트 의의, 활용방안, 자체 평가 결과

프로젝트는 GPU 운영환경을 구축해봤다는 점, 다양한 딥러닝 패키지 탐색하고 비교해 봤다는 점, 모델을 설계하며 공부를 해봤다는 점에 있어서 의의가 있다고 생각함.

01

GPU 운영환경 구축
운영환경은 최신 버전이 좋은 것이 아니라,
호환이 되는 버전이 중요하다는 점을 깨달았음.

02

다양한 딥러닝 패키지 탐색
OpenCv, Keras, face-recognition 등 **다양한
패키지들을 비교**하여 적절한 패키지를 사용했음

03

모델을 분석하며 공부
모델 input과 output의 shape에 대해 **모델을
분석하여 공부**한 것이 많은 도움이 되었음.



안면인식 기술은 더욱 높은 정확도와 새로운 기능의 추가를 바탕으로 여러 분야, 다양한 서비스에 접목되어 새로운 가치를 창출할 수 있다고 예상됨.



<상담/면접 시 감정 분석>

- 심리 및 정신 상담 시 **환자의 얼굴을 인식하고 분석**하여 참고 자료로 활용가능.
- 면접관 없이 진행하는 **온라인 시면접** 등에 활용가능.
(이미 시면접을 시행하는 기업이 존재)



<타겟 위치 알림>

- 사람을 대상으로 했을 경우, **미아**를 찾거나 행방 불명된 **취객 찾기** 등에 사용 가능 (적은 데이터로 학습 가능 시)
- 사물을 대상으로 했을 경우, 불량품의 위치 추적 등에 사용 가능



<백화점 VIP 고객 찾기>

- 백화점 특정 코너를 자주 방문하는 고객일 경우, 얼굴을 인식하여 **맞춤 상품 추천** 가능
- VIP 고객 방문 정보를 직원에게 전달하여 더욱 **친절한 응대**가 가능하게 함.



자체 평가 결과

01 주제설명

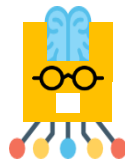
연구소장 주도 하에 자체평가 리스트 3가지를 각각 평가함. 각각의 평가 결과는
① 파이썬 구사능력 90점, ② 모델 적합성 82점, ③ 모델 최적화 95점이라고 도출됨.

구분		평가 근거	점수
① 파이썬 구사 능력	<ul style="list-style-type: none">파이썬을 활용하여 딥러닝을 위한 데이터 수집을 할 수 있는가?파이썬을 활용하여 모델에 맞는 데이터 정제를 할 수 있는가?	<ol style="list-style-type: none">파이썬으로 라벨/비라벨 데이터를 가져오는 코드를 구현함.전처리 프로세스와 기능을 파이썬으로 구현함.	94점
② 모델 적합성	<ul style="list-style-type: none">주제에 맞는 적절한 딥러닝 모델을 사용했는가?연구원 모두가 자신만의 모델을 개발했는가?모델의 타당성을 입증할 수 있는가?	<ol style="list-style-type: none">영상 딥러닝에 맞는 CNN 모델이 사용됨연구원 4명이 모두 자신만의 모델을 개발정확성을 근거로 모델 타당성을 증명함	82점
③ 모델 최적화	<ul style="list-style-type: none">최적모델을 선정하기 위한 과정이 잘 설명되었는가?파라미터 별 모델 평가를 충분히 하였는가?	<ol style="list-style-type: none">연구원 별 모델을 취합하여 최적의 모델을 개발하는 과정을 잘 설명함유의함 파라미터를 사용하여 정확도 향상	92점





Thank you



백선생 딥러닝 연구소