

AI 개발자 트랙 미니프로젝트 4차

# 얼굴 인식 (Face Recognition)

## AI 1반 1조

김효연, 박정은, 박지수, 유지현, 유현종, 조강윤, 최재영



# 1. FaceNet

## 1 데이터 및 데이터셋 구조

### 직접 찍은 데이터셋 (my)



### 데이터셋

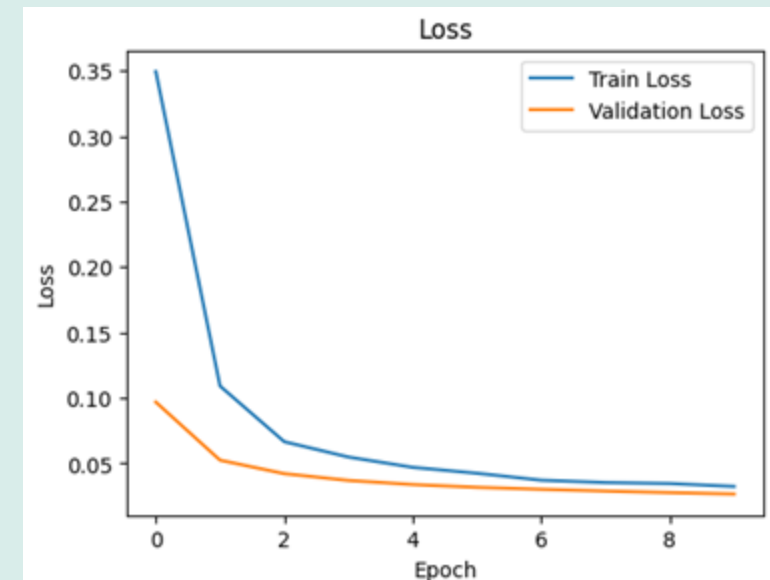
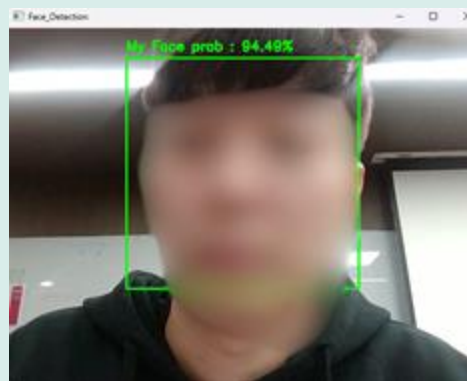
My face와 Other face의 비율을 맞춰서 학습

Training data의 my\_face 이미지 수 : 3600  
 Training data의 other\_face 이미지 수 : 3600  
 Test data의 my\_face 이미지 수 : 901  
 Test data의 other\_face 이미지 수 : 901

### 문제점

FaceNet특성상 얼굴 간 세밀한 차이를 인식함으로 얼굴을 가까이해야 잘 탐지

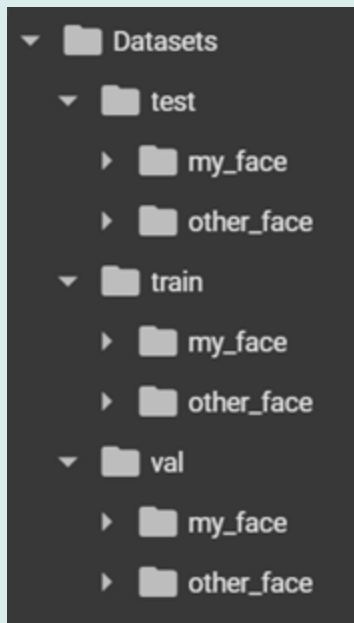
## 2 실행 결과



	precision	recall	f1-score	support
other	0.50	0.50	0.50	901
my	0.50	0.50	0.50	901
accuracy			0.50	1802
macro avg	0.50	0.50	0.50	1802
weighted avg	0.50	0.50	0.50	1802

## 2. YOLO-cls

### 1 데이터 및 데이터셋 구조



#### ▪ 직접 찍은 데이터셋 (my)



#### ▪ Haar Cascade로 얼굴 탐지

#### ▪ Data Augmentaion 사용:

좌우 반전, 회전, 확대/축소, 밝기 변화

#### ▪ 주어진 데이터셋 (other)



#### ▪ 학습에 사용된 데이터 개수 :

Training Images - Other: 7410, My: 1400  
Validation Images - Other: 3176, My: 600  
Test Images - Other: 2647, My: 500

### 2 사용 알고리즘 및 모델

#### ▪ Haar Cascade 알고리즘



#### ▪ UltraLytics YOLO-cls n 모델

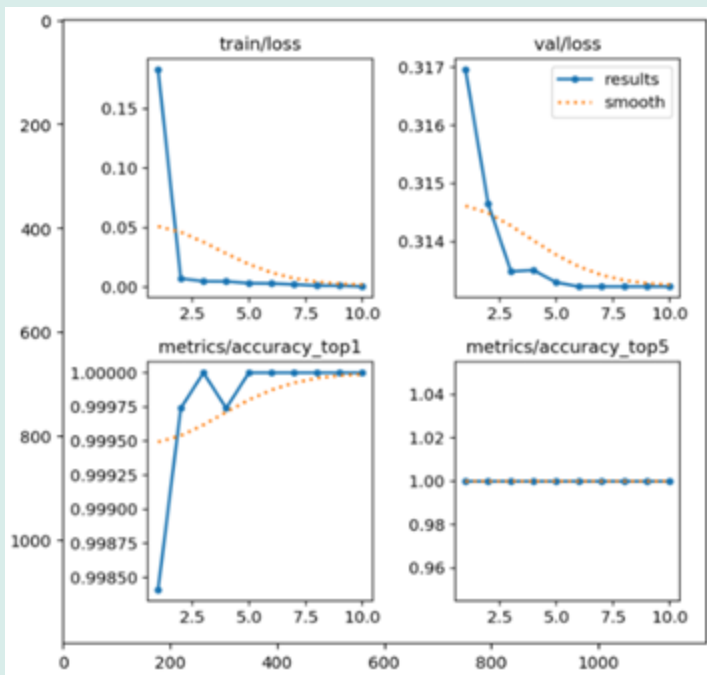
: n, s, m, l, x 중에서 가장 빠른 모델



## 2. YOLO-cls

### 3 학습

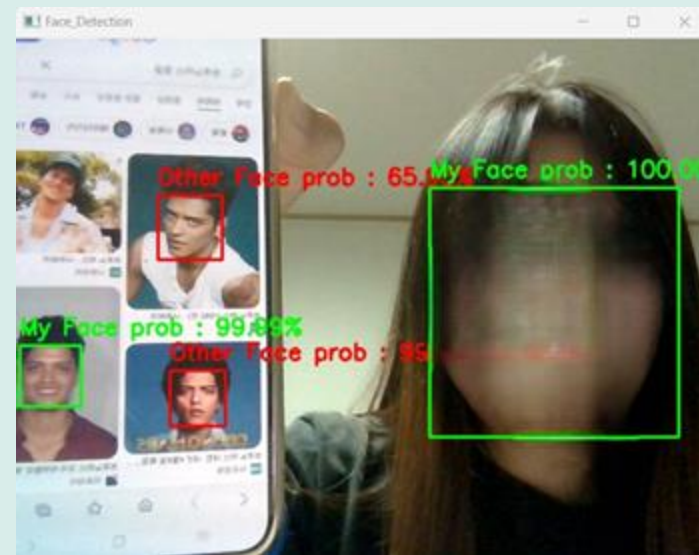
Epoch	GPU_mem	loss	Instances	Size
10/10	0.241G	0.0003718	106	64:



- Epochs : 10
- Img Size : 64
- Batch : 128

loss는 계속 낮아지면서 10번째 epoch에서 0.00037의 낮은 결과를 보여주었고, 높은 정확도로 학습됨

### 4 실행 결과



- 본인의 얼굴은 잘 인식함
- 타인의 얼굴은 항상 분류하지는 못하고, 내 얼굴로 인식하는 경우도 존재함

➡ 더 다양한 조건에서 이미지를 확보하여 학습하는 과정이 필요

# 3. YOLO : 1차

## 1 학습에 사용된 데이터

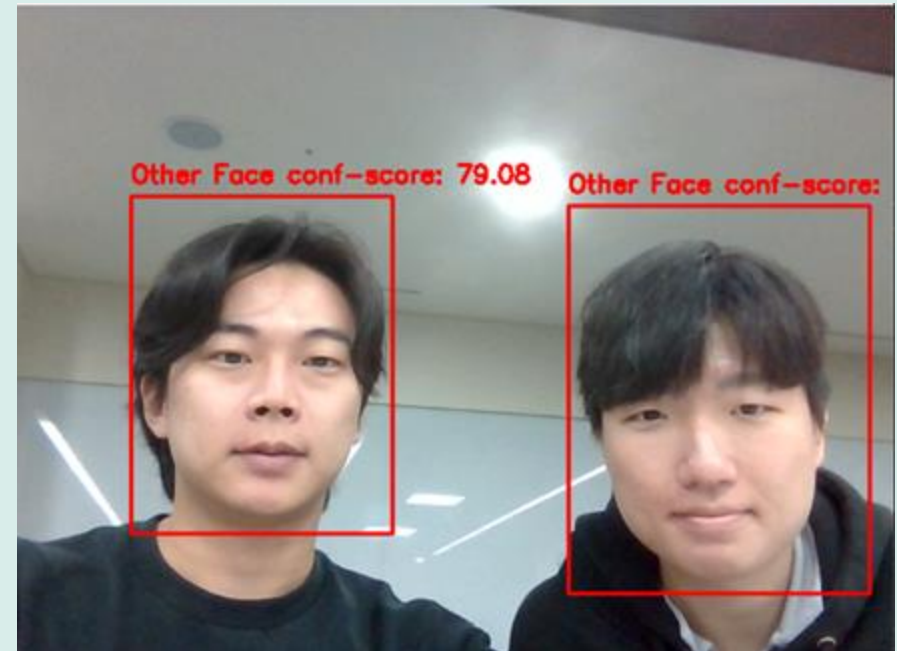


### Auto Labeling

- 얼굴이 탐지된 **크롭 이미지** 사용, 전체 이미지 크기가 얼굴로 인식하도록 라벨링 작업
- annotation = '1 0.5 0.5 1 1'

Class	Images	Instances	Box	R	mAP50	mAP50-90
all	2231	2590	0.954	0.881	0.921	0.838
others	1506	1865	0.915	0.763	0.847	0.701
my	725	725	0.993	1	0.995	0.975

## 2 실행 결과



✓ 결론 : 모든 데이터에 동일한 값을 일괄 적용한 것이 **Localization 학습에 악영향을 미쳐 성능 부진**



### 3. YOLO : 2차

#### 1 학습에 사용된 데이터

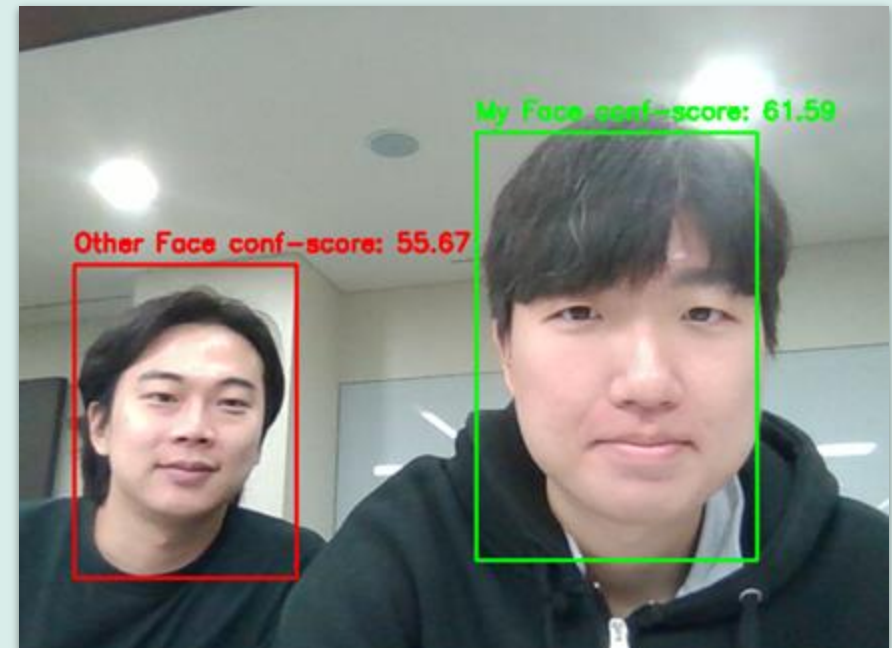


##### ■ Ybat Labeling

- **배경이 포함된 이미지**를 정사각형으로 resize
- Ybat 프로그램을 이용한 라벨링

Class	Images	Instances	Box	R	mAP50	mAP50-90
all	2006	2365	0.936	0.849	0.895	0.543
others	1506	1865	0.942	0.762	0.862	0.709
my	500	500	0.93	0.936	0.928	0.377

#### 2 실행 결과



✓ 결론 : Ybat을 이용한 수동 라벨링을 진행했을 때 **성능 개선**

# 3. YOLO : 3차

## 1 학습에 사용된 데이터

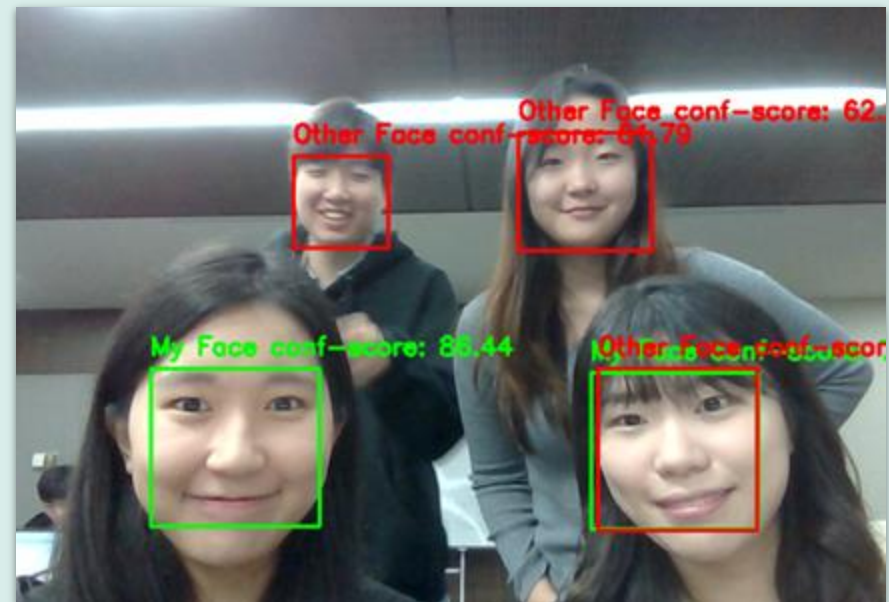


### ▪ Ybat Labeling

- 크롭된 이미지를 Ybat 프로그램 이용해 라벨링

✓ 결론 : 크롭된 이미지를 사용하더라도 **Box 라벨링** 작업이 정확할수록 모델 정확도 증가

## 2 실행 결과



## 4. 종합 결과

모델	주요 목적	데이터구조	특징	장점	단점
FaceNet	얼굴 인식	이미지	<ul style="list-style-type: none"> <li>얼굴의 임베딩 벡터화 및 유사도 기반 인식</li> <li>명암으로 특징 추출</li> </ul>	얼굴 간 세밀한 차이 인식 가능	<ul style="list-style-type: none"> <li>얼굴 인식에 한정적</li> <li>일반 객체에 부적합</li> </ul>
YOLO-cls	이미지 분류	이미지	<ul style="list-style-type: none"> <li>단일 단계의 클래스 분류</li> <li>명암으로 특징 추출</li> </ul>	높은 분류 성능과 빠른 속도	<ul style="list-style-type: none"> <li>위치 정보 부재</li> <li>탐지 작업에 부적합</li> </ul>
YOLO	객체 검출	이미지+라벨	<ul style="list-style-type: none"> <li>다중 객체 검출 및 분류, 위치 파악</li> <li>어노테이션 작업이 추가로 필요</li> </ul>	실시간 다중 객체 검출 가능	<ul style="list-style-type: none"> <li>작은 객체 구분 약함</li> <li>복잡한 배경에서 오탐 가능</li> </ul>

- ✓ 얼굴 인식 프로그램 개발 시에는 FaceNet, 제조업에서 불량인지 정상인지 분류하는 품질검사에는 YOLO-cls, 얼굴 세밀하게 추출할 필요 없이 다양한 객체 탐지가 필요한 자율주행 자동차에는 YOLO가 적합함
- ✓ 우리의 궁극적인 목적은 사내 출입 과정에서의 오랜 대기 시간을 줄이기 위함임으로 세밀한 얼굴 추출에 특화된 **FaceNet모델이 보다 적합**할 것으로 보임



