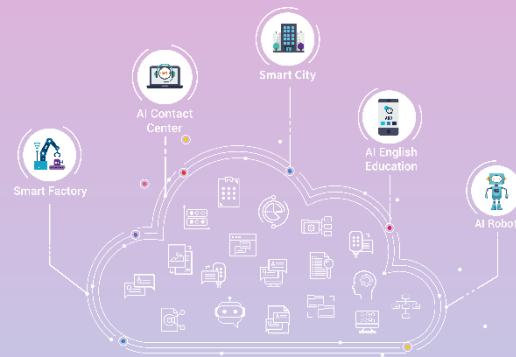


Level 2. How to customize AI engines



<http://mindslab.ai>

1. Face Recognition 구조

2. Face Recognition 알고리즘

3. Face Recognition 예제

4. Face Recognition 실습



Face Recognition 이란

- 주어진 사진에서
 - 1) 얼굴 부분을 **검출(Detect)**하고,
 - 2) 검출된 사진이 기존에 알고 있던 인물과 비슷한 지 **인식(recognition)**
- Open-set/Zero-shot 방식의 검출/인식을 사용
 - 별도의 추가 학습을 필요로 하지 않음

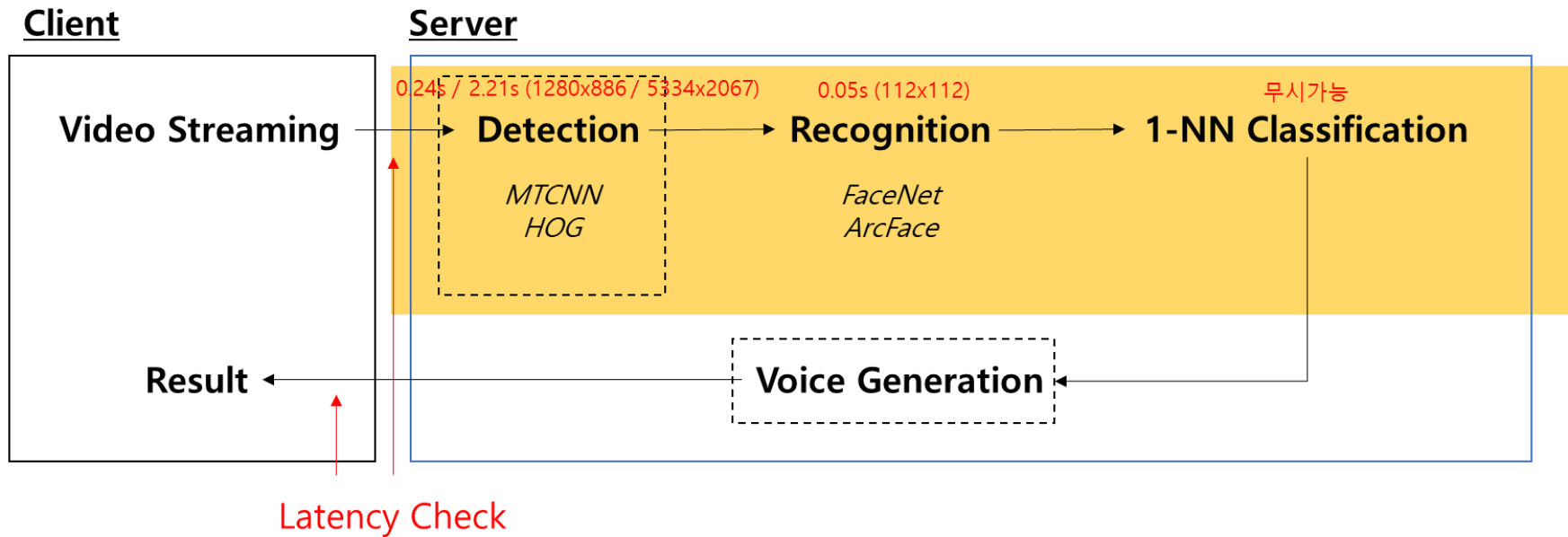
✓ **Input** : Video Stream

✓ **Output** : Identification Result

✓ **Data** : - (pivot data images conditionally)

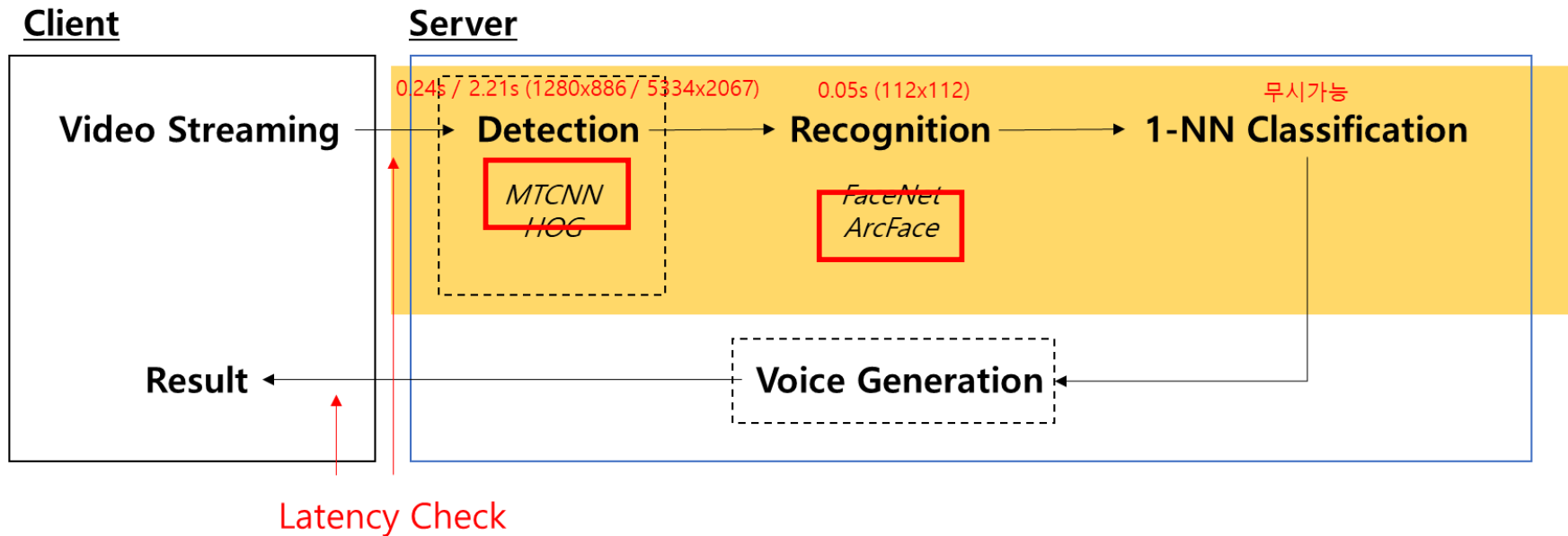
Face Recognition 구조

전체 Flow



Face Recognition 구조

전체 Flow



- Detection Model이 Front에서 제공되는 경우 Recognition 만 API 형태로 사용 가능
 - 5층 키오스크
- Latency/Edge device 제약에 따라 알고리즘 선택 가능

1. Face Recognition 구조

2. Face Recognition 알고리즘

3. Face Recognition 예제

4. Face Recognition 실습



✓ Closed Problem

- ✓ 이미 공개된 모델로 충분한 정확도를 보이는 것이 확인됨
- ✓ 특정 인물에 대한 최적화 필요 x

✓ BUT Real-time issue

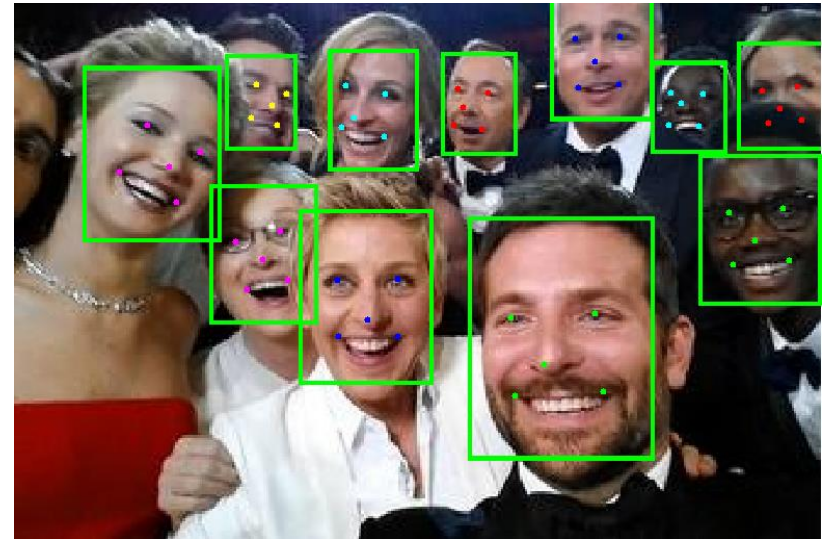
✓ 사진 화질 (앞 단의 latency와 연관됨)

✓ 연산 속도 vs. 성능

(HOG)

(MTCNN)

로컬 머신에서 충분히 가능
(라즈베리 파이도 가능!)



Face Recognition 알고리즘

Recognition

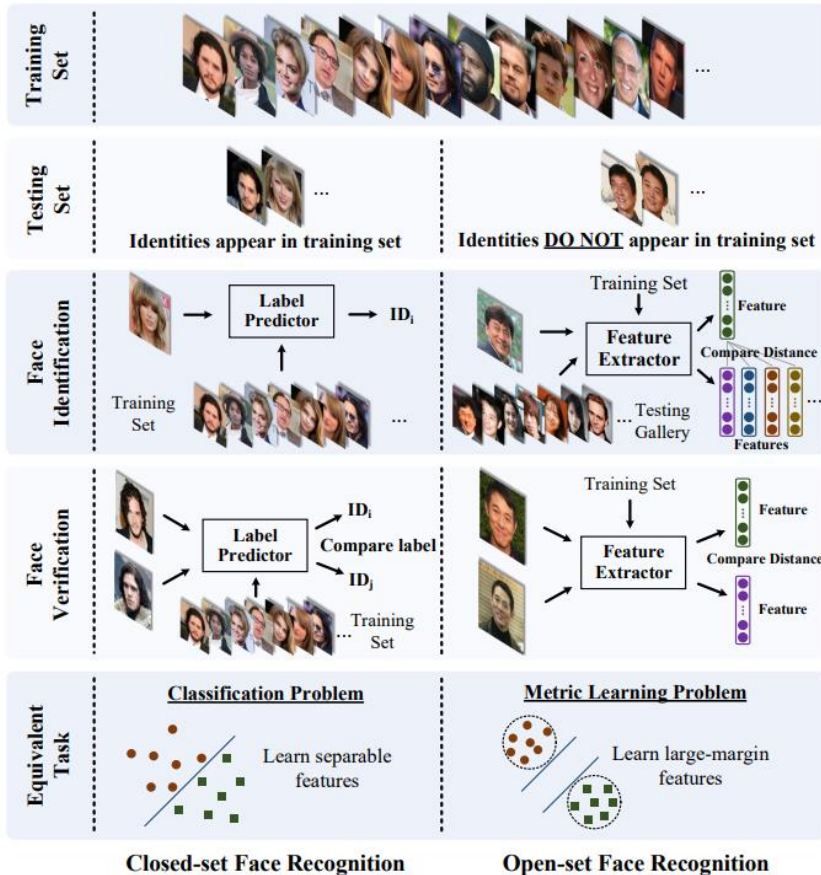


Figure 1: Comparison of open-set and closed-set face recognition.

- Closed-set vs. **Open-set**
 - 유저 추가시마다 학습 새로 하는 것이 아님
 - 추가 학습 필요치 않아 매우 효율적
 - **Zero-shot learning**
- Identification/Verification Problem 모두에 적용 가능

Face Recognition 알고리즘

Recognition

- 현재 Face Recognition의 SOTA 알고리즘은 ArcFace
 - 99.4% 가량의 정확도를 보여줌
- 기존 FaceNet 방식에 비하여, 구별하기 힘든 사람들(Hard Negative)을 보다 잘 구별할 수 있도록 Loss 설계

Who's Who?

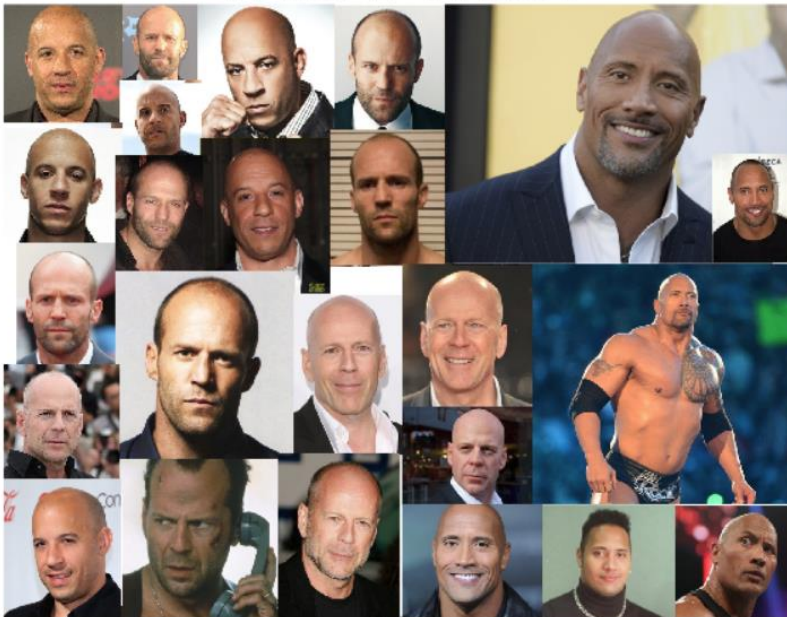


Figure 1

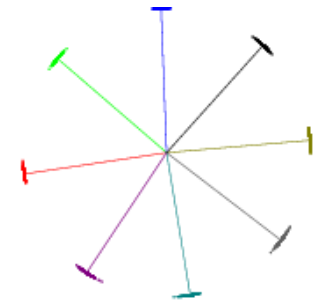


Figure 2

Face Recognition 알고리즘

Recognition

- 현재 Face Recognition의 SOTA 알고리즘은 ArcFace
 - 99.4% 가량의 정확도를 보여줌
- 기존 FaceNet 방식에 비하여, 구별하기 힘든 사람들(Hard Negative)을 보다 잘 구별할 수 있도록 Loss 설계

FaceNet(2015)

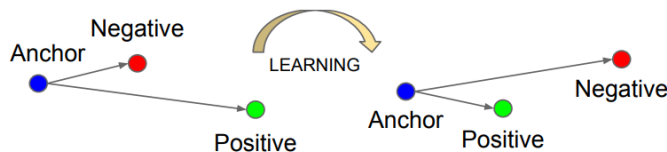


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

• ArcFace(2018)

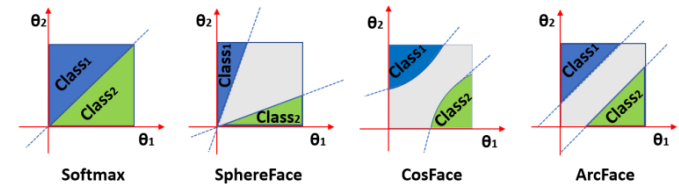


Figure 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

1. Face Recognition 구조

2. Face Recognition 알고리즘

3. Face Recognition 예제

4. Face Recognition 실습



Face Recognition 예제

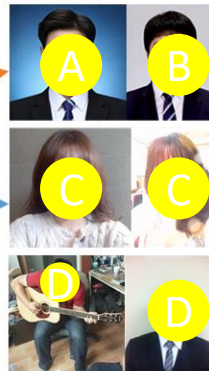
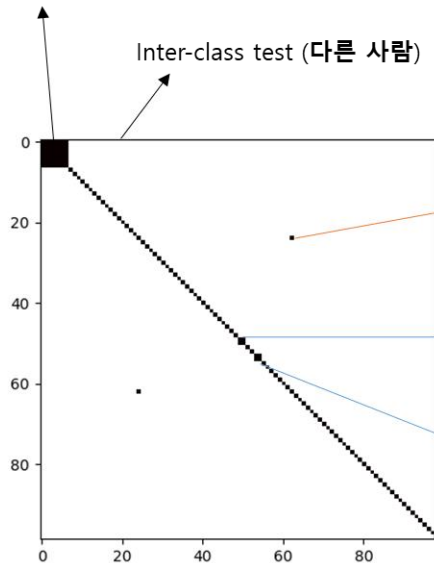
Mindlab 사원 HeatMap

- 사내직원 100분 slack/docswave 사진 + 테스트용 동일 사진 7장 테스트
- Heatmap에서 자신이 다른 사람들과 얼마나 동일한지 여부를 확인할 수 있음 (100x100)
 - 약 5000번의 비교에서 1번 오검출
 - 사원 데이터에 학습시키지 않음

Intra-class test (같은 사람)

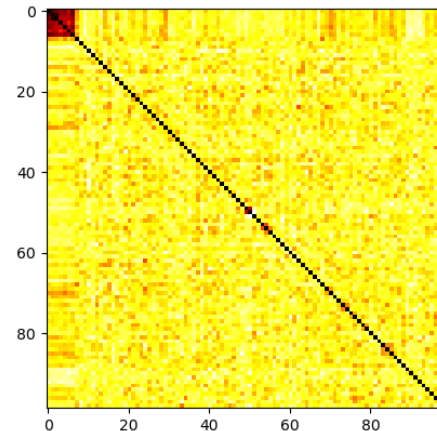


Inter-class test (다른 사람)



다른 사람이나 오탐

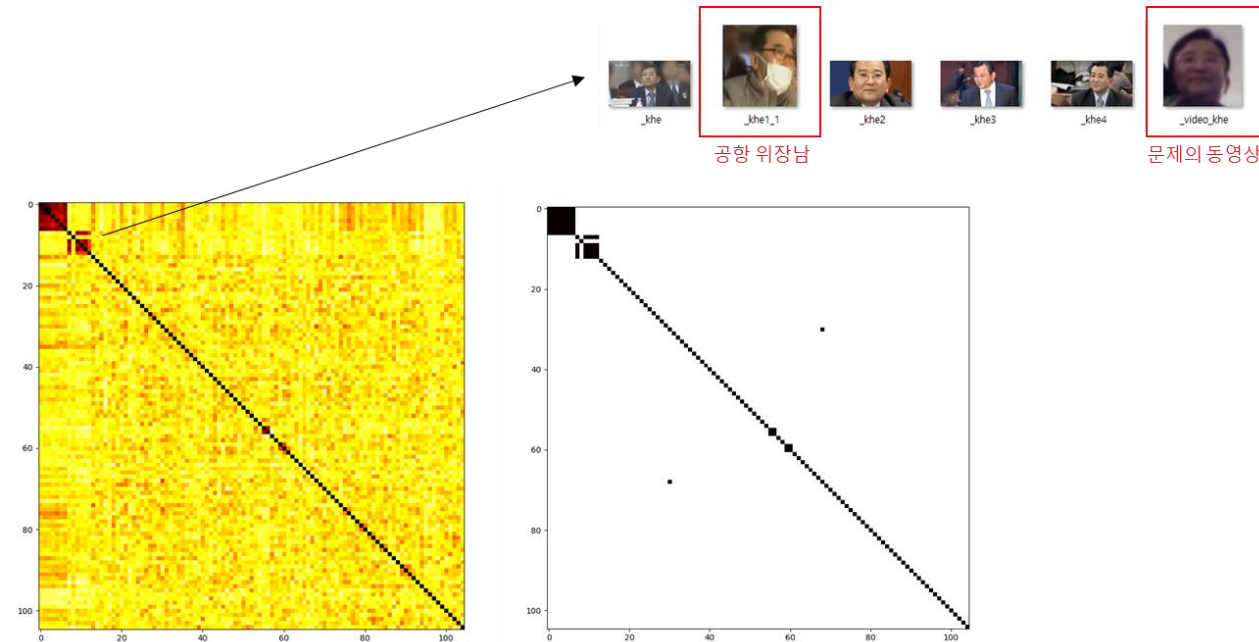
같은 사람 찾음



Face Recognition 예제

Mindlab 사원 HeatMap

- 사내직원 100분 slack/docswave 사진 + 테스트용 동일 사진 7장 테스트
- Heatmap에서 자신이 다른 사람들과 얼마나 동일한지 여부를 확인할 수 있음 (100x100)
 - 약 5000번의 비교에서 1번 오검출
 - 사원 데이터에 학습시키지 않음



1. Face Recognition 구조

2. Face Recognition 알고리즘

3. Face Recognition 예제

4. Face Recognition 실습



Face Recognition 실습

기본 설정

- <https://github.com/mindslab-ai/brain-fr>
 - 권한 필요
 - ./deploy/test.py 를 중점으로 기본적인 text 가능
 - Detection / Recognition / 다른 이미지(pivot image)와의 비교 / Heatmap 그리기 / 나이나 성별 추정하기 등

```
def get_features_from_one_image(img_path):  
    img = cv2.imread(img_path)  
    img = model.get_input(img)  
    if img is None:  
        print('No face detected in %s' % img_path)  
        img_ft=None  
    else:  
        img_ft = model.get_feature(img)  
  
    return img_ft
```

```
def get_features_from_images(input_dir_path):  
    img_path_list = []  
  
    full_list = os.listdir(input_dir_path)  
    full_list.sort()  
  
    for name in full_list:  
        if (name.endswith('.png') or name.endswith('.jpg') or name.endswith('.PNG') or name.endswith('.JPG')):  
            img_path_list.append(os.path.join(input_dir_path, name))  
  
    img_features = []  
    print("[img_path_list]")  
    for img_path in img_path_list:  
        print(img_path)  
        img_ft = get_features_from_one_image(img_path)  
  
        #print(img_ft[0:10])  
        img_features.append(img_ft)  
  
        ## For age tracking,  
        # gender, age = model.get_ga(img)  
        # print(gender)  
        # print(age)  
  
    return img_features
```

Face Recognition 실습

기본 설정

- <https://github.com/mindslab-ai/brain-fr>
 - 세부 옵션 설정 가능
 - pivot_root : 비교 대상 image folder
 - input_root: input image folder
 - threshold(구분 기준): default 1.24로 되어있으나, precision이나 recall의 중요도에 따라 약간씩 변형 가능(default 추천)
 - gpu: 사용유무 설정 가능
 - 기타 세부 옵션들 역시 python ./deploy/test.py -help 명령어를 통해 확인할 수 있음

```
parser.add_argument('--threshold', default=1.24, type=float, help='ver dist threshold')
parser.add_argument('--input_root', default='../pivot', help='input data root path')
parser.add_argument('--pivot_root', default='./face_db.json', help='pivot data root path')
parser.add_argument('--tmp_folder', default='./tmp', help='tmp image location')
```


Face Recognition 실습

Grpc server/client

- 각 모델에 대한 수정이 필요한 경우 ./face_model.py 수정
- 현재는 grpc를 사용하여 wrapping 되어 있기 때문에 api 형태로 직접 vector 값을 호출 받을 수 있음
 - ./face_server.py ./face_client.py

```
if __name__ == "__main__":  
    args = parse_args()  
    insight_server = InsightFaceServer(args)  
    server = grpc.server(futures.ThreadPoolExecutor(max_workers=1),)  
    insight_servicer = add_InsightFaceServicer_to_server(insight_server, server)  
    server.add_insecure_port('[::]:{}'.format(args.port))  
    server.start()  
  
    print("insight face starting at 0.0.0.0:{}".format(args.port))  
  
    try:  
        while True:  
            time.sleep(60 * 60 * 24)  
    except KeyboardInterrupt:  
        server.stop(0)
```

관련 자료



- <https://github.com/deepinsight/insightface>
- Liu, Weiyang, et al. "Sphereface: Deep hypersphere embedding for face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017
- <https://arxiv.org/abs/1801.07698>

감사합니다.

MINDs Lab. AI Platform Company