

---

# Real-time pedestrian pose estimation, tracking and localization for social distancing

---

20230000 김병훈

# CONTENTS

1. Introduction
2. Related works
3. Methods
4. Experiments
5. Conclusion

# 1. Introduction

## Pose Estimation

- Pose Estimation이란?
  - 이미지나 비디오에서 Object의 Semantic Key points를 검출하고, Key points간의 관련성을 통해 위치를 추정하는 문제
    - ✓ Semantic Key point란, Object에서 의미가 있는 값
    - ✓ Human을 대상으로 Pose estimation을 할 경우에는 "오른쪽 어깨", "왼쪽 무릎" 등의 값이 Key point 값으로 검출된다.
  - 방향성이나 위치를 감지(Detecting)하고, 연결(Associating), 추적(Tracking)하는 것을 포함하기 때문에, 컴퓨터 비전 분야의 Basic Task라고 할 수 있음.
- 활용 분야

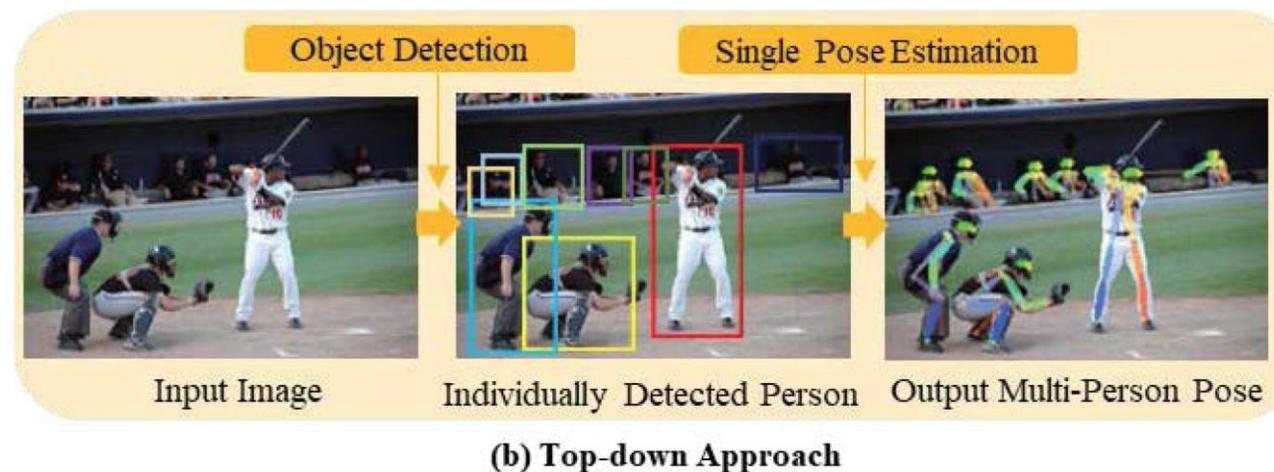
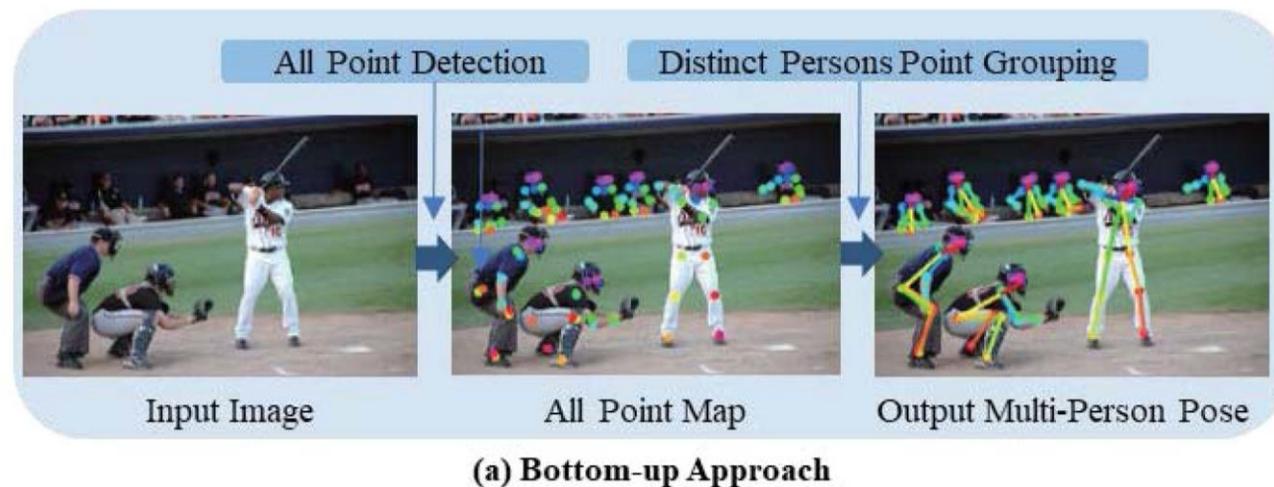


# 1. Introduction

## Human Pose Estimation

- Pose Estimation의 접근법

- 1) Bottom-up method
- 2) Top-down method



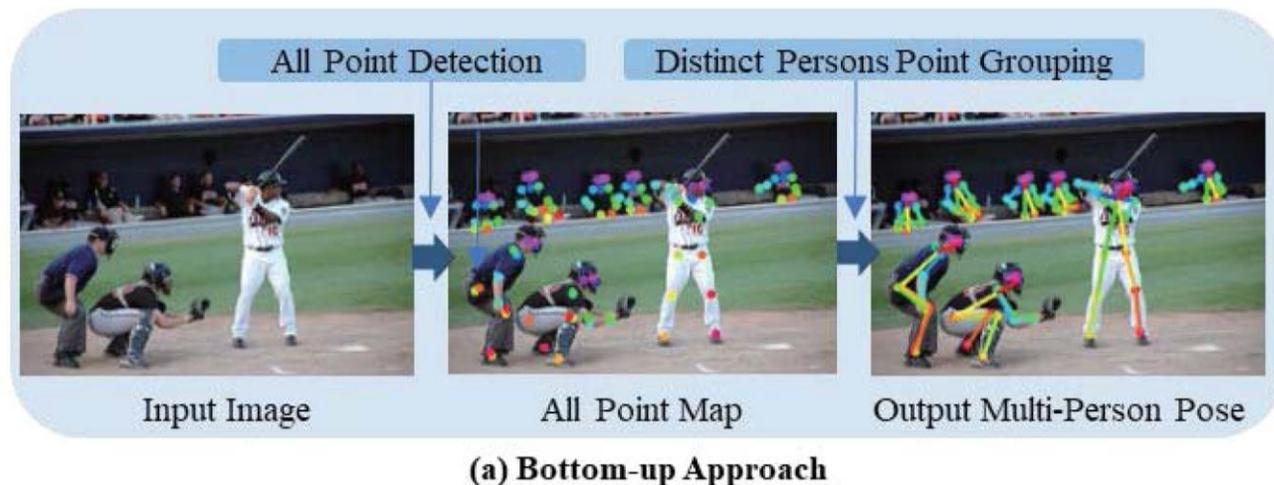
# 1. Introduction

## Human Pose Estimation

- Pose Estimation의 접근법

- 1) Bottom-up method

1. 영상에 나타나는 Key points를 모두 추정
2. Key points간의 상관관계를 분석
3. 포즈 추정

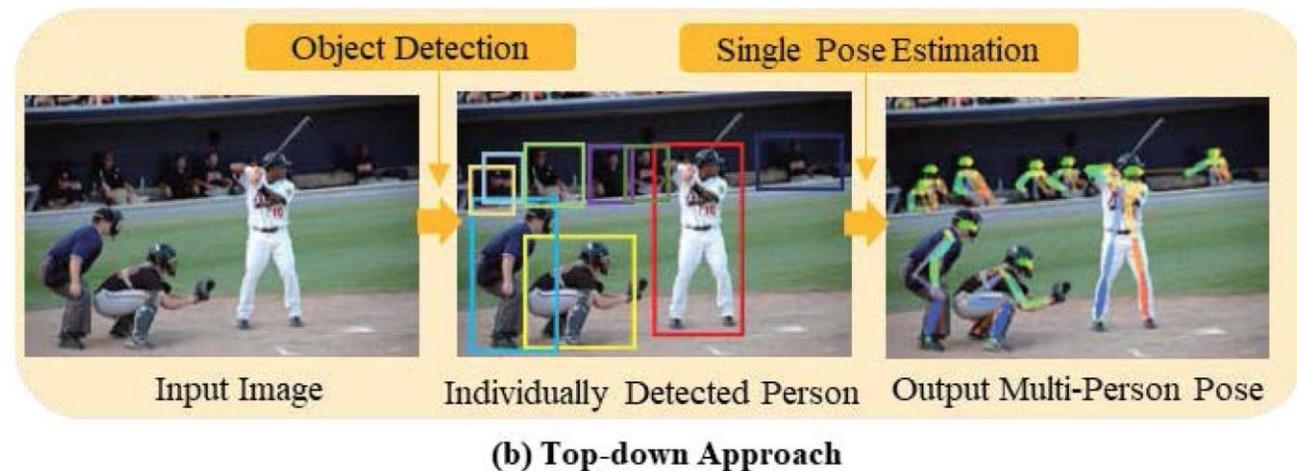


- ✓ 별도의 Detection과 Cropping 과정이 필요 없음
- ✓ 위 과정을 건너뛴으로써, 연산 속도가 빠름 (Multi-Person 포즈 추정에 적합)
- ✓ 하지만 Key points를 적절하게 매칭하는 것이 어렵기 때문에 정확도가 Top-down 방식에 비해 낮음

# 1. Introduction

## Human Pose Estimation

- Pose Estimation의 접근법
  - 2) Top-down method
    1. Object Detection(Bounding box)
    2. 신체 관절(key points)을 매칭
    3. 관절 정보를 통해 자세 추정

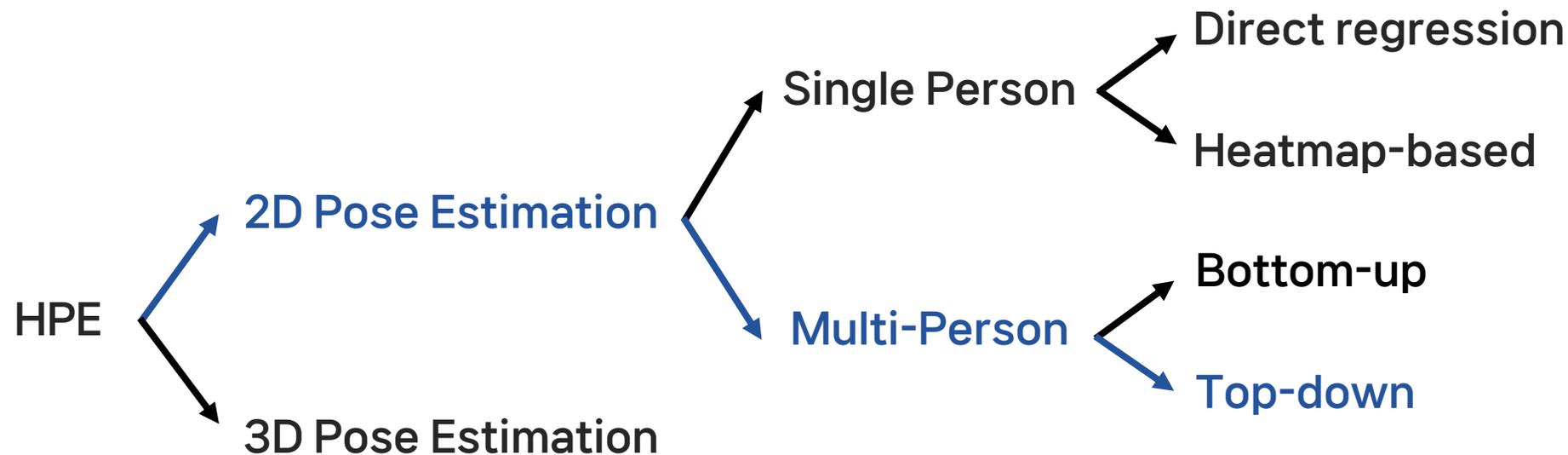


- ✓ 별도의 Detection과 Cropping 과정이 필요, 대표적 모델 Mask-R-CNN(2017, CVPR)
- ✓ 전체 이미지에서 관절을 추정하는 Bottom-up 방식보다 정확도가 높음
- ✓ 사람을 인식하지 못하면 추정을 할 수 없음
- ✓ Multi-Person 의 포즈를 추정할 때, 계산량의 증가로 속도가 느린 문제

# 1. Introduction

## Human Pose Estimation

- Real-time pedestrian pose estimation, tracking and localization for social distancing (22.12.05)



# 1. Introduction

Real-time pedestrian pose estimation, tracking and localization for social distancing (22.12.05)

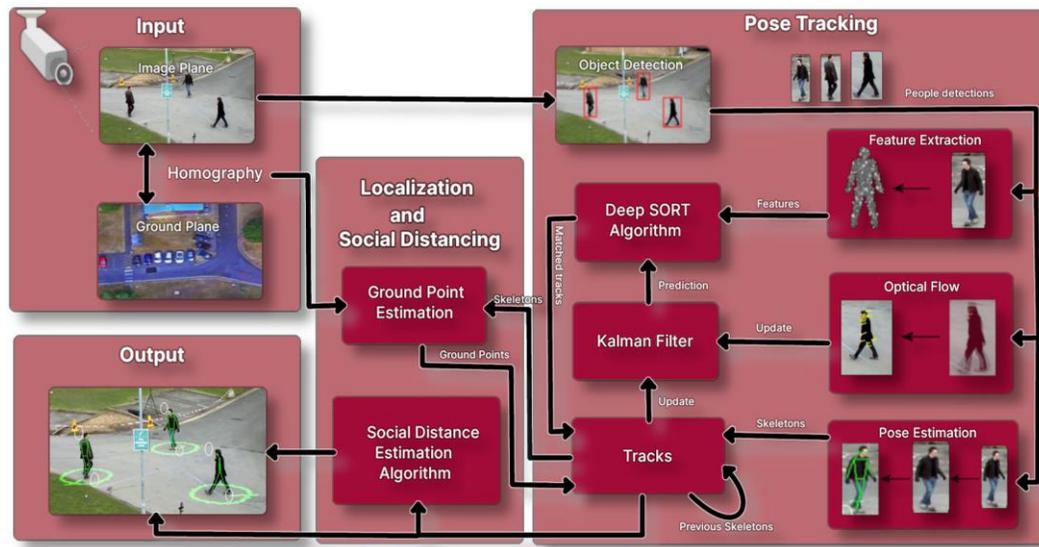


Fig 1. Workflow of the PETL4SD system pipeline

## PETL4SD system

1. Pedestrian Detection
2. Pose estimation
3. Feature extraction
4. Tracking
5. Localization
6. Social distancing calculation
7. Visualization of the tracks and social distance violations

- ✓ COVID-19 팬데믹 상황 속, 미국 질병통제예방센터(CDC)는 사회적 거리 두기 정책을 시행
- ✓ 개인 간 안전 거리로 2M 권장
- ✓ 사회적 거리 두기를 측정할 위해 PETL4SD system 제안

# 1. Introduction

Real-time pedestrian pose estimation, tracking and localization for social distancing (22.12.05)

- Contribution

1. 자세 추정에 대해 하향식 접근 방식의 모델을 상향식으로 구현하는 방법 사용
  - ✓ 자세 추정 결과 개선 / 경량화된 모델을 사용하여 낮은 개발환경에서도 실시간으로 작동 할 수 있음.
2. 사람의 몸에 대한 평균 분포를 사용하고, Tracking 기록을 이용하여 Pose 추정
  - ✓ Occlusion으로 인해 발을 감지하지 못해도 발의 위치를 추정
  - ✓ 이전의 데이터를 활용하여 합리적인 추정치를 얻으므로 위치 정확도를 향상
3. 스마트 위반 감지기 구축
  - ✓ 보행자들 간의 거리, 자세 추정을 통해 사회적 거리 두기 위반을 감지

---

정확성과 효율성 측면에서 기존 연구보다 더 높은 성능을 가진 PETL4SD System 제안

## 2. Related works

### 2.1. Related works tables

Paper	Year	Contribution
Reaserch and implementation of social distancing monitoring technology based on SSD [28]	2021	SSD 아키텍처, 사회적 거리두기 계산
A deep learning-based social distance monitoring framework for COVID-19 [2]	2021	YOLOv3, 사회적 거리두기 계산
Social Distance Monitoring Approach Using Wearable Smart Tags [3]	2021	스마트 웨어러블 태그
Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques [27]	2020	YOLOv3, DeepSORT, 사회적 거리두기 위반 지수 계산
Simple online and realtime tracking with a deep association metric [34]	2017	DeepSORT 알고리즘
Using Computer Vision to enhance Safety of Workforce in Manufacturing in a Post COVID World [26]	2020	근로자 활동 모니터링
A Vision-based Social Distancing and Critical Density Detection System for COVID-19 [35]	2020	음향 신호를 활용한 경고 시스템
The Visual Social Distancing Problem [12]	2020	Pose estimation, 객체 거리 측정
Omni-Scale Feature Learning for Person Re-Identification [1]	2019	Pose estimation, 객체 거리 측정
MonoLoco: Monocular 3D Pedestrian Localization and Uncertainty Estimation [7]	2019	Pose estimation, 객체 거리 측정

Solution	Year	Contribution
Landing AI Creates an AI Tool to Help Customers Monitor Social Distancing in the Workplace [21]	2020	CCTV를 사용한 거리두기 감지기
Social Distance Monitoring [16]	2020	LiDAR 기반 시스템
Using 3D Cameras to Monitor Social Distancing [31]	2020	스테레오 카메라 기반 시스템

## 2. Related works

### 2.2. Related works Limitation

- Limitation

1. 정확한 추적과 위치 추정을 위한 Pose 추정은 이루어지지 않음
2. B-box를 통해 위치 추정을 수행하여, 실제 위치에 대한 모호한 추정
3. 추가적인 하드웨어를 필요로 하여 비용이 발생
4. 개인정보의 문제
5. 가족이나 친인척에 대한 그룹화 과정이 생략되어, 잘못된 사회적 거리두기 위반을 추정하는 경우가 발생

# 3. Methods

## 3.1. Planar homography estimation

- Homography

- 정의 : 3차원 공간상의 평면을 서로 다른 시점에서 바라봤을 때 획득되는 영상 사이의 관계
- 두 평면 간의 4개의 대응점을 찾아서 행렬 계산을 실시
- 대응점을 설정할 수 없는 경우에는 싹 기하학 또는 기계 학습을 통해 호모그래피 추정이 가능
  - ✓ 하지만, 신뢰성의 문제가 있음(정확도 측면)

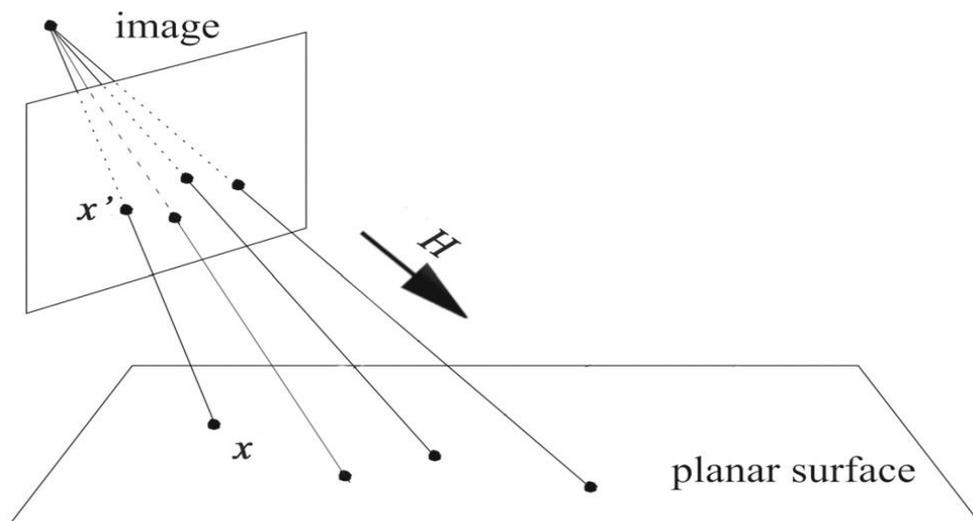
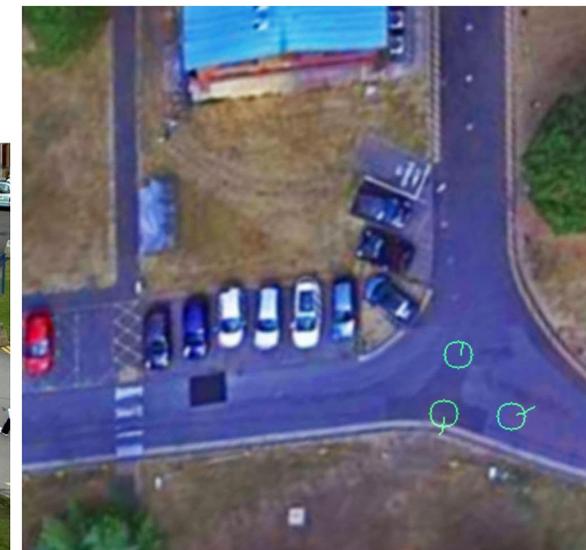


Figure 2. A visual representation of homography and the corresponding points



(a)



(b)

# 3. Methods

## 3.2. Pedestrian detection with YOLOv5

- YOLOv5
  - 물체 감지를 위한 빠르고 정확한 신경망
    - ✓ 입력 이미지를 그리드로 나누고, 각 그리드에 대해 경계 상자와 클래스 확률을 생성
  - 테스트 시에 전체 이미지를 고려하기 때문에 예측 결과가 이미지의 전체적인 컨텍스트에 영향을 받음.
  - 하나의 네트워크 평가로 예측을 수행하므로 빠름(Fast R-CNN보다 100배)

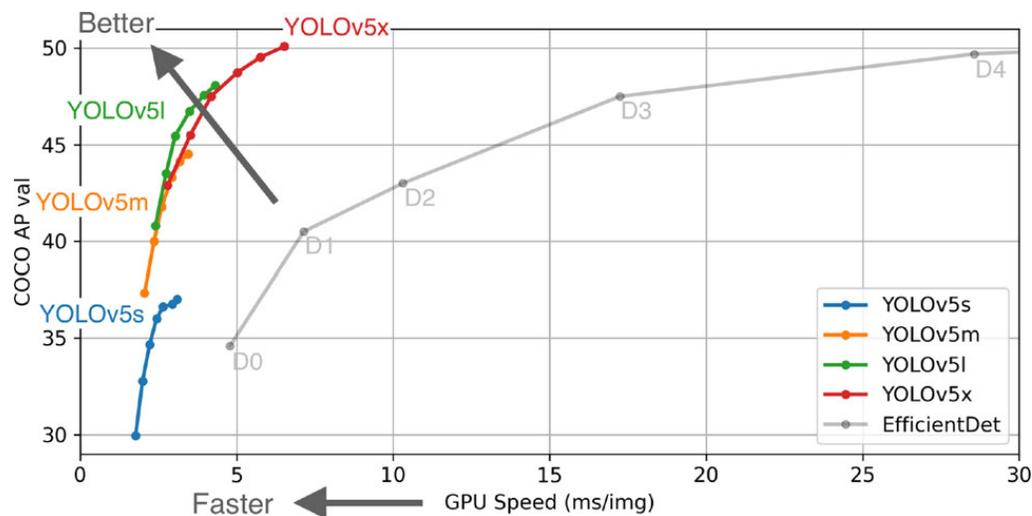


Figure 3. Graph showcasing performance of different YOLOv5 models taken from YOLOv5 GitHub repository

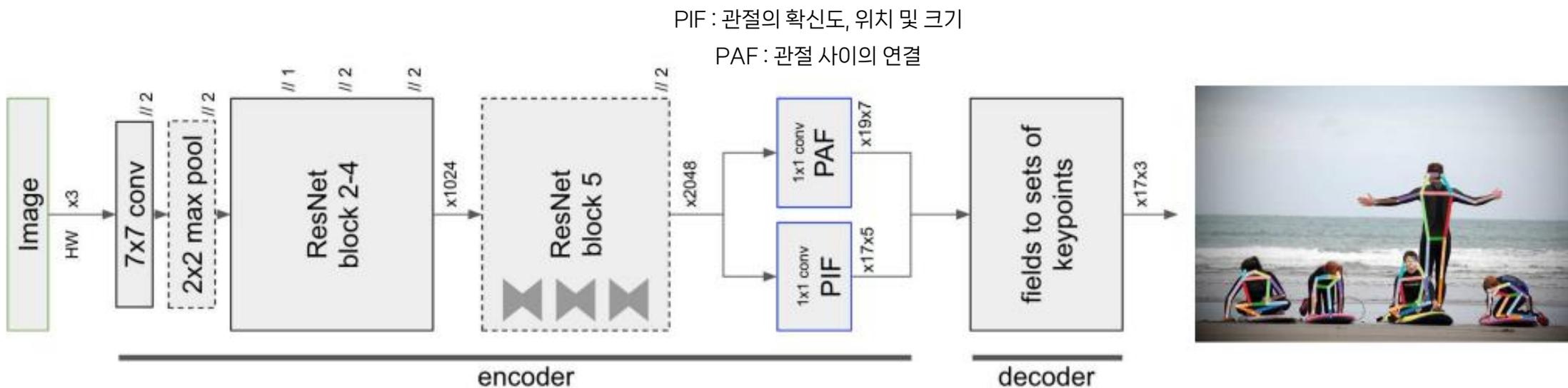


Figure 4. Bounding boxes drawn on an original frame from the result of the YOLOv5 neural net

# 3. Methods

## 3.3. Human pose estimation using OpenPifPaf with detection

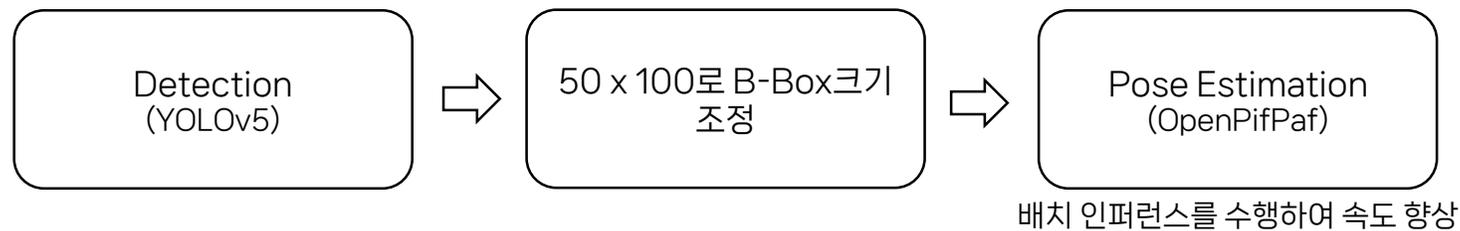
- OpenPifPaf
  - COCO Datasets으로 사전 훈련된 모델 사용
  - Bottom up 방식으로 구동
  - 저해상도와 혼잡/가려짐으로 인한 추정이 어려운 데이터에 적합



## 3. Methods

### 3.3. Human pose estimation using OpenPifPaf with detection

- YOLOv5 + OpenPifPaf



**Table 2** Validation results on 100 pedestrian setting images from the CrowdPose dataset [22], after revising the validation set to include missed labels

Method	Average precision (AP)			Average recall (AR)			FPS
	IoU=0.50:0.95	IoU=0.75	IoU=0.50	IoU=0.50:0.95	IoU=0.75	IoU=0.50	
OpenPifPaf (solo)	0.421	0.451	0.625	0.573	0.613	0.722	2.85
Our method (with YOLOv5x)	0.451	0.470	<b>0.652</b>	<b>0.644</b>	0.676	<b>0.797</b>	<b>2.87</b>
Our method (with YOLOv5l)	0.441	0.467	0.633	0.638	0.677	0.789	3.45
Our method (with YOLOv5m)	0.458	0.491	<b>0.667</b>	0.640	<b>0.680</b>	0.792	4
Our method (with YOLOv5s)	0.455	0.493	<b>0.650</b>	0.623	0.669	0.762	<b>5</b>
DEKR	<b>0.515</b>	<b>0.585</b>	0.652	0.593	0.644	0.680	1.66

## 3. Methods

### 3.3. Human pose estimation using OpenPifPaf with detection

- YOLOv5 + OpenPifPaf



Figure 8. A comparison from the CrowdPose dataset showcasing superior performance of our method on far away pedestrians: a shows output from our method (YOLOv5 bounding box and OpenPifPaf pose estimation); b shows output from DEKR [14]

# 3. Methods

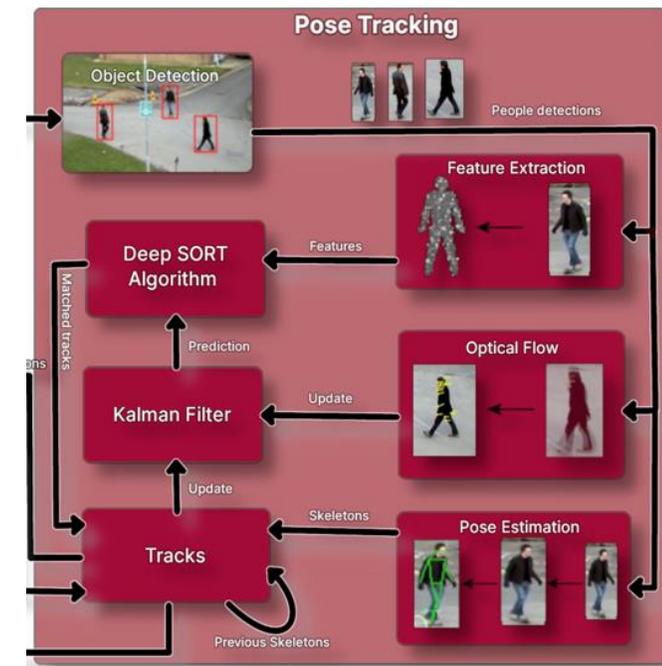
## 3.4. Tracking using DeepSORT with optical flow

- DeepSORT 알고리즘 작동 방식

1. 객체 감지 및 추적 초기화 : 입력된 영상에서 객체를 감지하고 b-box를 찾음
2. 특징 추출 : 추적할 객체마다 특징을 추출(객체를 고유하기 식별하기 위해)
3. 트랙 유지 및 업데이트 : 이전 프레임에서 추적된 객체의 트랙 정보를 유지하고, 현재 프레임에서 새로운 위치정보를 업데이트
4. 트랙의 연결 및 분리 : 다음 프레임에서 이전의 정보를 활용하여 감지된 객체들을 매칭

- 본 논문의 제안 추적 시스템

- 특징 추출기 신경망을 OSNET REID로 대체 : 경량화 모델 사용으로 실시간 추론 가능
- **Optical Flow**를 특징으로 추가 : 추적 정확도와 카메라 이동 감지를 향상
- Optical Flow를 예측하여, 객체 감지 과정을 건너 뛴 (계산 비용 줄임, 성능 향상)
- 객체 감지기를 YOLOv5로 대체하여 처리 속도 향상
- 카메라를 정지상태로 가정하여 카메라의 이동을 감지하는 부분을 없앴(불필요한 계산 줄임)



## 3. Methods

### 3.4. Tracking using DeepSORT with optical flow

- Optical flow 추출

1. 그레이 스케일로 변환
2. Shi-Tomasi의 코너 감지 방법 사용 : 타원체 필터를 통과하여 강한 코너 지점을 추출
3. 포인트 필터링 : B-box를 중심으로 타원 영역 내에 없는 키 포인트를 삭제, 이전에 예측된 키포인트가 있으면 재활용
4. 이전에 찾은 키포인트와 현재 프레임의 키포인트 사이를 매칭 (최대 오차 임계값 100을 넘어가는 포인트는 삭제)
5. 부분 어파인 변환 행렬 계산 : 현재 경계 상자를 Optical Flow로 예측된 위치로 변환
6. 칼만 필터 업데이트 : 중간 과정에서는 Optical Flow로 예측한 위치로 업데이트 / 최종 추적이 완료된 상태에서 재 업데이트
7. B-box 매칭 및 추적 : Mahalanobis 거리와 특징 임베딩 간의 유클리드 거리를 사용하여 매칭
8. 트랙 매칭 및 재식별

## 3. Methods

### 3.5. Pedestrian localization and distance calculation

- 1) 가려짐이 없는 경우
  - 두 발 사이의 중심으로 추정
- 2) 다리의 일부만 검출된 경우
  - 우측 수식을 이용하여 추정
- 3) 다리 전체가 가려진 경우
  - 평균 인체 비율 분포에 기반하여 추정
- 4) 바운딩 박스만 검출된 경우(스켈레톤 검출 X)
  - Tracking 기록으로 평균 보행자 크기 추정
  - B-Box의 하단 중심으로 추정

$$F = \left( x_t + d \cdot \sqrt{\frac{1}{1+m^2}}, y_t + d \cdot m \cdot \sqrt{\frac{1}{1+m^2}} \right)$$

$m$  : 허벅지의 기울기,  $x_t, y_t$  : 무릎의 좌표,  $d$  : 종아리의 길이



# 3. Methods

## 3.6. Smart violations detector

- 보행자 그룹 감지
  - 1) 걷는 경우 : 프레셰 거리를 계산
  - 2) 서 있는 경우 : 서로를 바라보고 있는 경우에만 그룹으로 판단



# 3. Methods

## 3.6. Smart violations detector

- 보행자 그룹 감지
  - 1) 걷는 경우 : 프레셰 거리를 계산
    - ✓ 프레셰 거리 : 곡선들 사이의 유사성을 측정
    - ✓ 특정 임계값보다 작으면 두 보행자가 같이 걷는 것으로 판단



Figure 12. People walking and having Fréchet distance of their paths less than a threshold are considered together and have blue circles around them

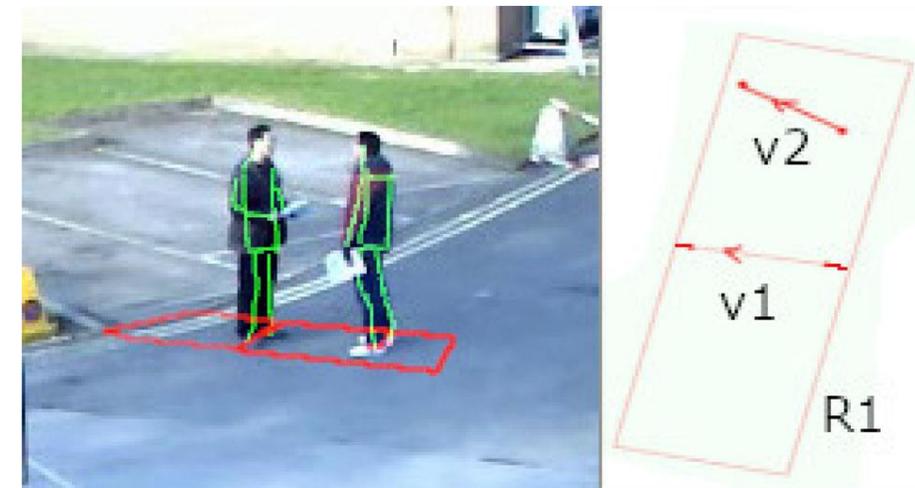
# 3. Methods

## 3.6. Smart violations detector

- 보행자 그룹 감지

2) 서 있는 경우 : 서로를 바라보고 있는 경우에만 그룹으로 판단

- ✓ 보행자 발의 좌표를 사용해서 1.5m의 직사각형을 그림.
  - ✓ 좌, 우측 발 좌표 사이에서 벡터  $v_1$ 을 구함
  - ✓  $v_1$ 의 기울기를 사용하여 직사각형  $R_1$ 을 그림( $V_1$ 의 너비 ==  $R_1$ 의 너비)
- ✓ 가까운 보행자의 발이 직사각형 내부에 있는지 or 벡터가 직사각형에 교차하는지 확인
- ✓ 두 벡터( $v_1, v_2$ )의 각도가 90도에 가깝지 않은지 확인
  - 한명은 카메라를 정면으로 바라보고, 한명은 카메라를 등진 경우
  - 측면에서 서로 바라보는 경우



# 3. Methods

## 3.6. Smart violations detector

- 보행자 그룹 감지

2) 서 있는 경우 : 서로를 바라보고 있는 경우에만 그룹으로 판단

- 두 보행자의 눈(Occlusion이 발생할 경우를 대비해 어깨/귀 포함)을 Detection
  - $e_{l1}$ ,  $e_{r1}$ ,  $e_{l2}$ ,  $e_{r2}$ 라는 Boolean 변수 사용( $ch_1$ ,  $ch_2$ )
  - 두 보행자 모두의 눈이 감지되거나 둘 다 감지되지 않을 경우
    - 두 보행자의 눈 간의 거리와 귀 간의 거리를 측정
    - 눈과 눈의 거리가 눈과 귀의 거리보다 작을 경우 -> 바라보고 있음
    - 반대의 경우 -> 바라보고 있지 않음

---

**Algorithm 1** Check if two people are facing each other when standing

---

**Require:**  $v_2$  is inside or intersecting  $R_1$

**Ensure:**  $\theta_1$  is not close to 90

```

if any( $e_{l1}, e_{r1}$ ) and (not all( $e_{l2}, e_{r2}$ ) and  $ch_2$ ) then
  return True
else if any( $e_{l2}, e_{r2}$ ) and (not all( $e_{l1}, e_{r1}$ ) and  $ch_1$ ) then
  return True
else if any( $e_{l2}, e_{r2}$ ) and any( $e_{l1}, e_{r1}$ ) and (not all( $e_{l2}, e_{r2}$ ) or not
all( $e_{l1}, e_{r1}$ )) then
  if  $\theta_2 > 90$  and dist( $eye_1, ear_2$ ) > dist( $eye_1, eye_2$ ) then  $\triangleright \theta_2$  check
is optional
    return True
  end if
end if
Return False

```

---

## 3. Methods

### 3.6. Smart violations detector

- 보행자 그룹 감지(결과)
  - 데이터 셋 : Oxford Town Centre Dataset(옥스퍼드 시내 CCTV 데이터)
  - 보행자의 그룹에 대한 라벨링이 없기 때문에, 서로 상호작용(손을 잡거나, 대화)하는 경우에만 묶어서 Ground truth 확인

**Table 4** Total together detections from the Oxford Town Centre dataset [5] that are interacting with each other

Ground truth (GT)	Total detected	In GT	Not in GT
21	19	17	2

81%의 정확도

# 3. Methods

## 3.7. Social distancing visualization

- 사회적 거리두기 시각화 방법

1. 보행자들 사이의 거리가 5m 이내일 경우에 녹색 거리선을 추가하고 선 위에 거리를 표시하는 방법(2m 이내는 빨간색)

이 방법은 많은 보행자들이 보이는 경우 직관적이지 못하다는 단점

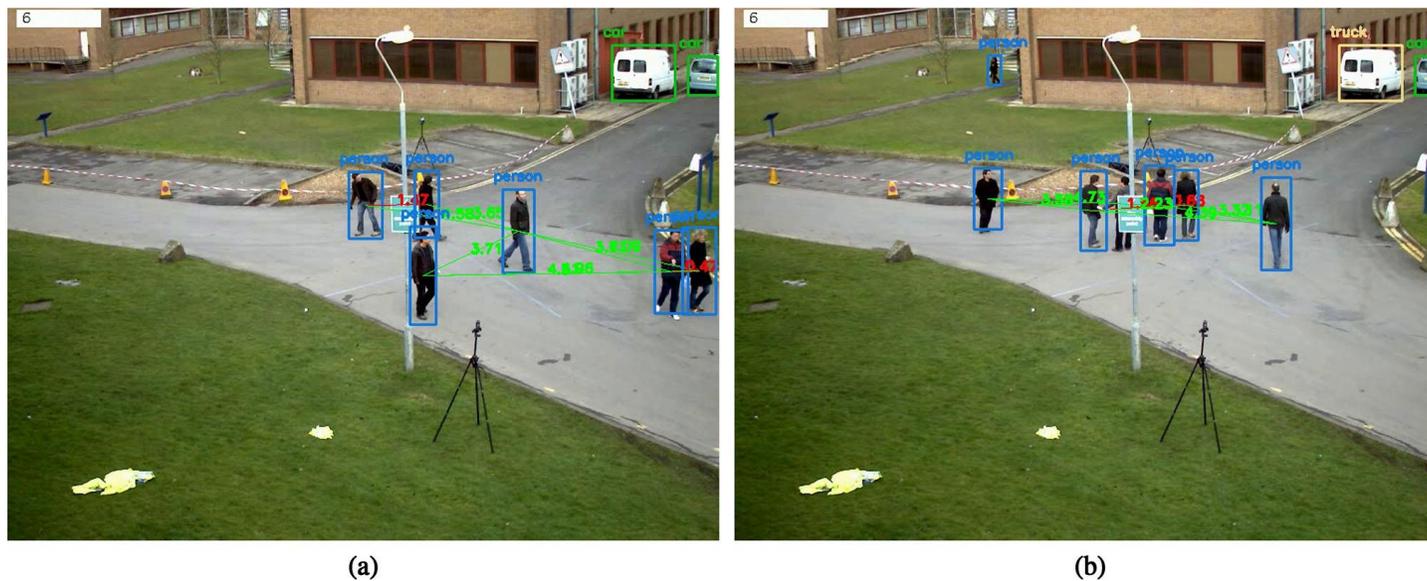


Figure 18. Social distancing initial representation with bounding boxes and lines: a Bounding boxes for pedestrians and lines between them in green ( greater than 6 feet) and red ( less than 6 feet). b Dense crowds may overwhelm the viewer in this kind of view

## 3. Methods

### 3.7. Social distancing visualization

- 사회적 거리두기 시각화 방법

2. 각 보행자 주변에 반지름이 3피트인 녹색 원을 그림(그룹은 파란색), 두 원이 겹치면 빨간색으로 변하며 경고

하지만 원의 크기를 지정할 경우 실제 거리를 정확하게 표시하지 못함

또한 실시간으로 원의 점을 변환 시키면 계산의 비효율성이 발생

별도의 빈 이미지에 Buffer를 추가하여 호모그래피 변환을 통해 실시간으로 원의 스케일을 변환

# 3. Methods

## 3.7. Social distancing visualization

- 사회적 거리두기 시각화 방법

$$P_i = H_{ip} \cdot P_p$$

$$P_b = H_{bp} \cdot P_p$$

$$P_p = H_{bp}^{-1} \cdot P_b$$

$$P_i = H_{ip} \cdot H_{bp}^{-1} \cdot P_b$$

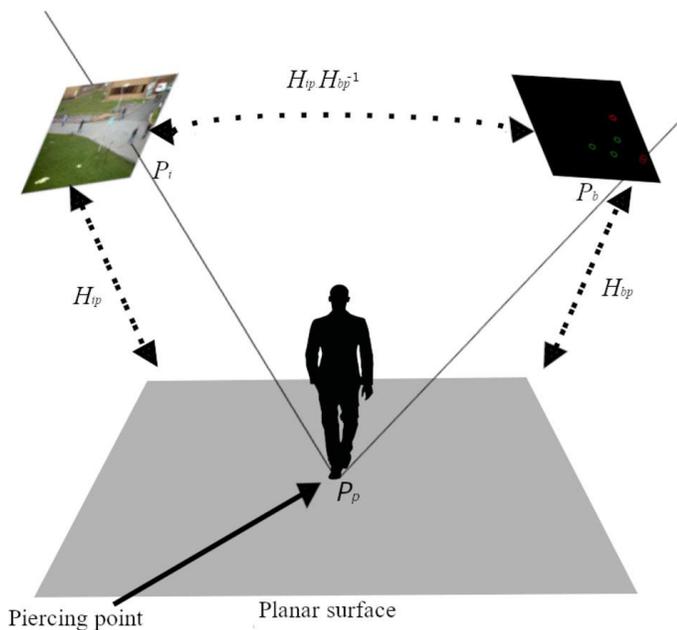


Figure 20. The homography between two images induced by a world plane (the concatenation of two homography matrices)

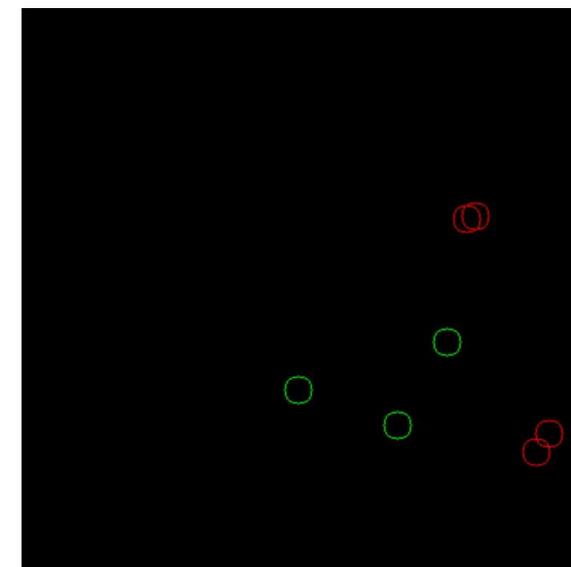


Figure 19. Scaled blank image top-down view

# 3. Methods

## 3.7. Social distancing visualization

- 사회적 거리두기 시각화 결과



(a)



(b)

Figure 21. 1 Social distancing final representation. a Social distancing enhanced visualization with circles around pedestrians. The image is a real-time composition of background, circles and pedestrian images(왼쪽) showing the right perspectives, sizes and occlusions. b Top down view with the social distancing circles added to the corresponding satellite image(오른쪽)

# 4. Experiments

## Experimental Settings

- Development Environment
  - CPU : Intel core i7-8750H
  - GPU : NVIDIA GTX 1050tiM
  - RAM : 16GB

Table 5. CPU, GPU, RAM usage and fps reported on the Oxford Town Centre dataset

Component	Average	Max
CPU utilization	44%	72%
GPU utilization	18.48% (747MB)	18.48% (747MB)
RAM utilization	2GB	2GB
FPS	5	7
Pedestrians visible in frame	15	24

The specifications of the machine are GPU:GTX 1050tiM and CPU:core i7-8750H

## 5. Conclusion

- CCTV 카메라를 사용하여 보행자의 자세를 실시간으로 추적하고 시각화 하는 PETL4SD 시스템 제안
- Pretrained 모델만 사용하여 추가적인 학습 없이 뛰어난 자세 추정 성능
- 보행자의 그룹을 지정하는 새로운 그룹화 기법 제공
- 최소 요구 환경이 낮아도 구현 가능

### 한계점

- Tracking과 성능을 향상하기 위해 카메라의 여러 각도를 추가하여 Occlusion 처리가 필요
- Panoptic segmentation과 평균 속도를 추정하는 방법을 통해 사람의 개입을 최소화 해야 함