



[졸업작품전 - 작품]

Photo Spot Manager at Edge platform

- 이한희 : 융합전자공학부 2019046108
- 김상협 : 융합전자공학부 2021051685



1. Motivation and Objectives

- 유명 관광지 방문 시 불편 사항

1. 포토 스팷을 한 사람이 너무 오랫동안 사용하는 문제
2. 새치기로 인한 혼잡 문제

'줄서기' 즐기는 MZ세대 그들은 왜?...불황에 '소확행·득템력' 열광

SNS 속 맛집·핫플레이스에 일상이 된 '줄 서기'

부산시, '새치기 NO, 줄서기 YES' 기초질서 캠페인 펼친다

줄서기 시비로 흥기 휘두른 50대 구속



1. Motivation and Objectives

- Solution : Photo Spot Management by **AI Platform!**



Video input



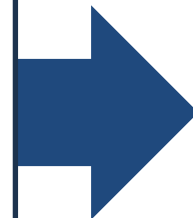
Detecting & Analyzing



Visualization



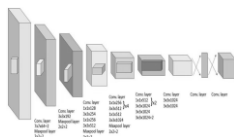
Alert using Speakers



1. Motivation and Objectives

- Edge AI Platform 구축할 때 **Challenges**

실시간 객체감지



안정적인 Tracking



연산성능의 제한

NVIDIA® Jetson Nano™

새치기 판단 및 경고



2. 실시간 객체 감지

- **Fine-tune** YOLO.v7 tiny model with 'People Detection' Dataset.

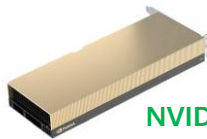
Class Balance

all train valid test

total: 17401



YOLOv7
tiny



NVIDIA L4

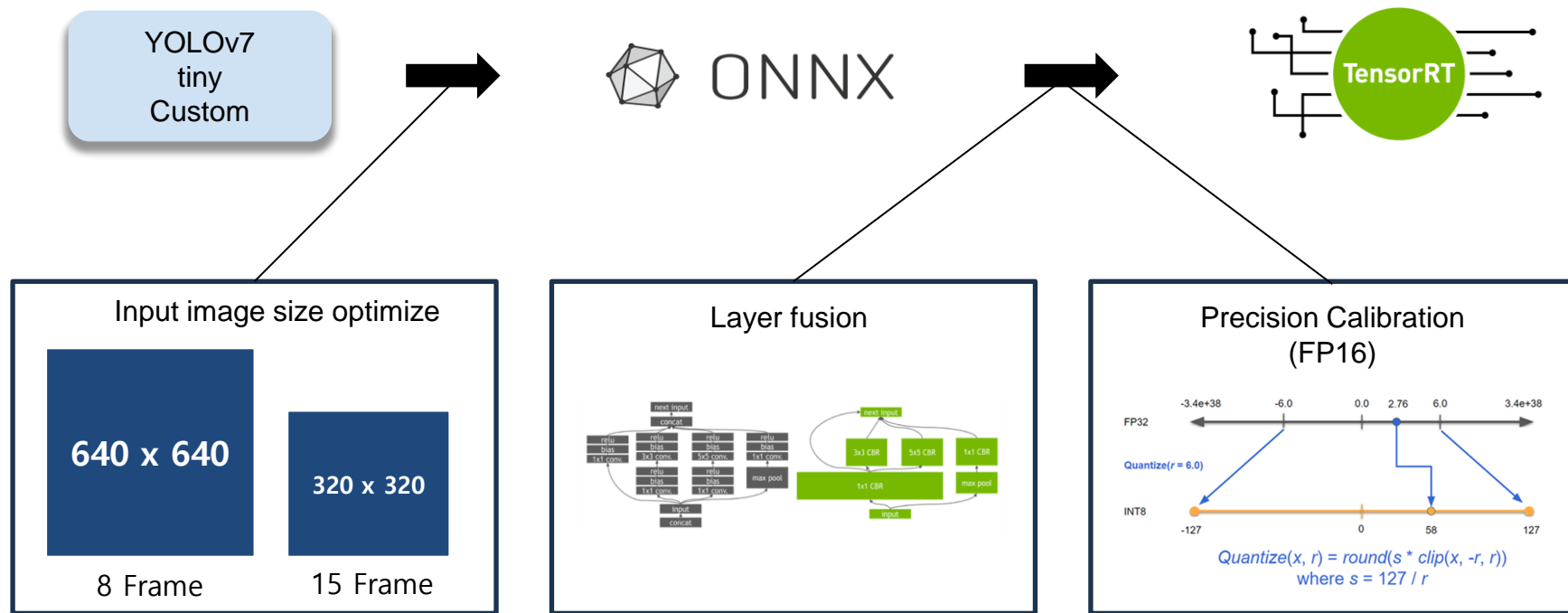


Batch 16
Epoch 55

YOLOv7
tiny
Custom

3. 연산 성능의 제한

- Optimization : Build **TensorRT** engine



4. 안정적인 Tracking

- 객체간 ID매칭

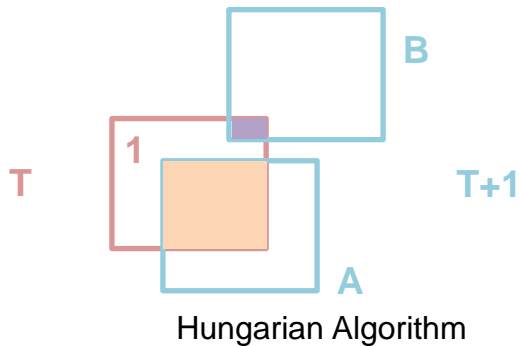


Tracker: ID가 부여된 박스

New Box: 아직 ID가 부여되지 않은 박스

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IOU



New box ID?

기존ID받기

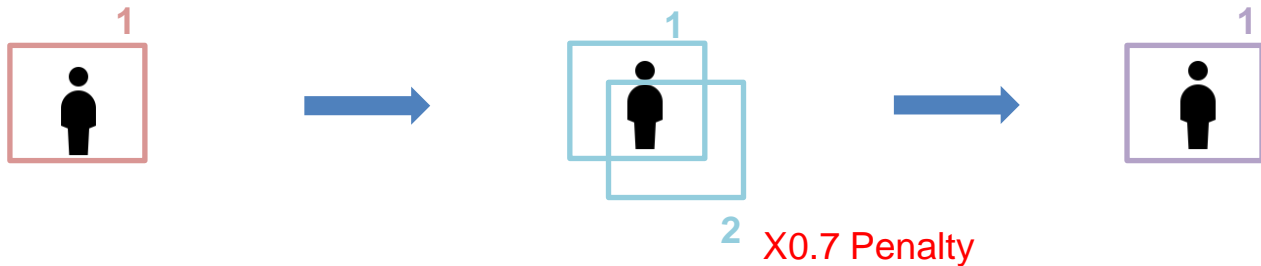
새로운ID받기

4. 안정적인 Tracking

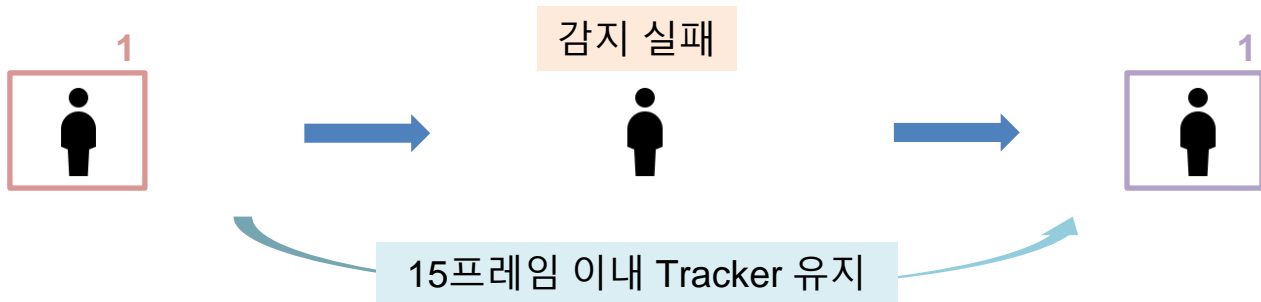
- Problem of ID confusing

Tracker: ID가 부여된 박스
New Box: 아직 ID가 부여되지 않은 박스

A. 하나의 객체에 여러 개의 박스가 반환되는 경우

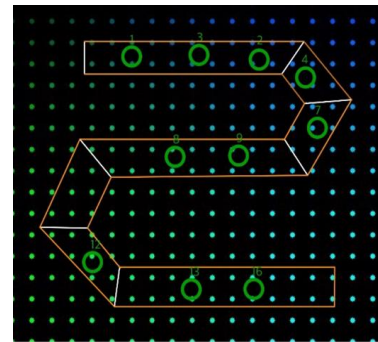
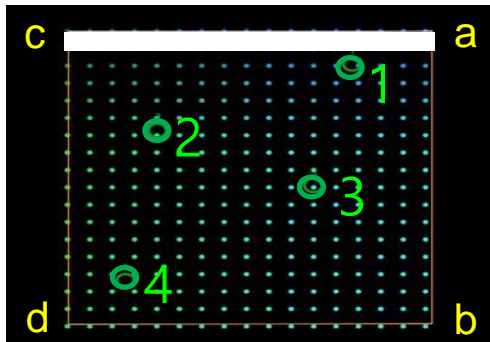


B. 객체의 박스가 반환되지 않는 경우

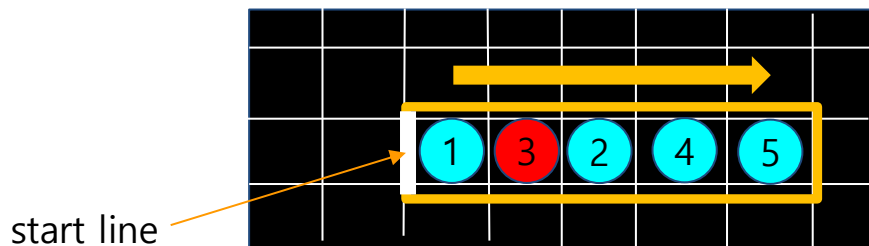


5. 새치기 판단 및 경고

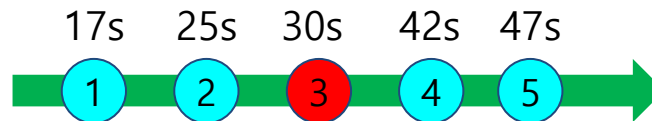
- 줄 안에 있는지 여부 & 순서 파악



- 새치기 감지



Location Order



Time Order

- 1. 마우스 클릭을 통한 영역설정

전체 감지 영역 설정
시간을 제한하는 포토 스팟 설정
사람들이 줄 서 있는 영역 설정

- 2. 포토 스팟에서 한 사람이 시간(20s) 초과할 경우
→ 붉은 색 원 으로 경고
 - 3. 새치기가 발생할 경우
→ 스피커를 통한 경고
-

- Articles

https://www.newsis.com/view/?id=NISX20181015_0000442490&cID=10811&pID=10800

<https://www.ggilbo.com/news/articleView.html?idxno=539987>

<https://www.news1.kr/society/general-society/4917580>

- Projects

https://developer.nvidia.com/embedded/community/jetson-projects/people_flow_analysis

https://developer.nvidia.com/embedded/community/jetson-projects/tracking_counting_people

- YOLO

<https://github.com/WongKinYiu/yolov7>

<https://blog.roboflow.com/yolov7-custom-dataset-training-tutorial/>

<https://universe.roboflow.com/leo-ueno/people-detection-o4rdr>

<https://eehoeskrap.tistory.com/414>

Q & A



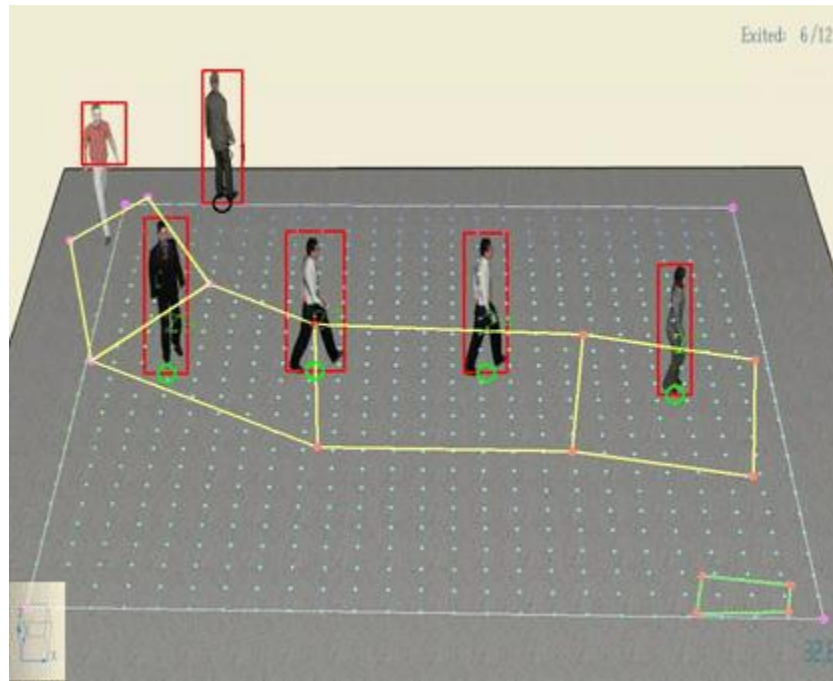
- 4. Model Test by **Crowd Simulation**



Difficulties in obtaining
human footage



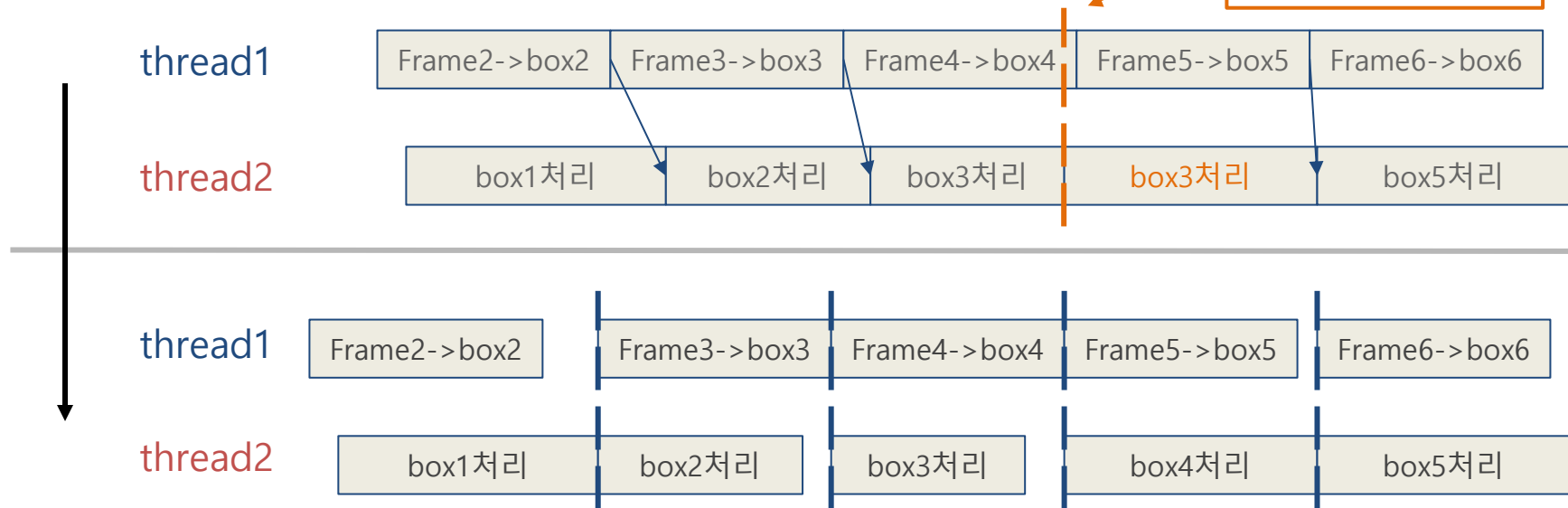
Used a simulator instead



- Multithreading : 동시에 실행되는 두가지 작업 동기화

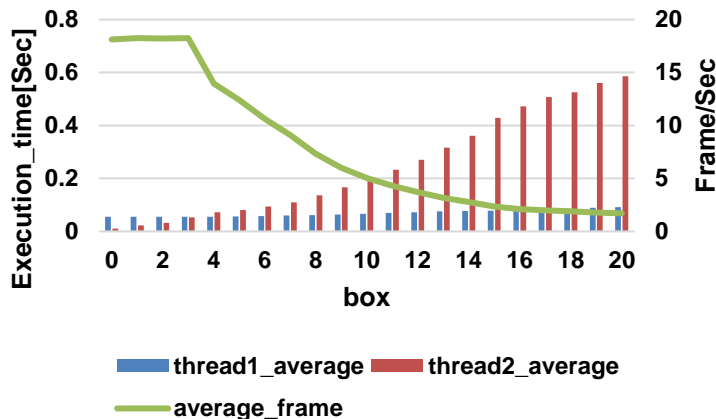
- GPU: YOLO v7 Object Detection (객체 좌표 반환)
- CPU: CV2 Library(화면 표시) + Cut-in-line Detection + ID tracking(좌표->객체)

thread1의 처리가 아직 끝나지 않았을때 thread2가 동작할경우 같은 작업 반복실행

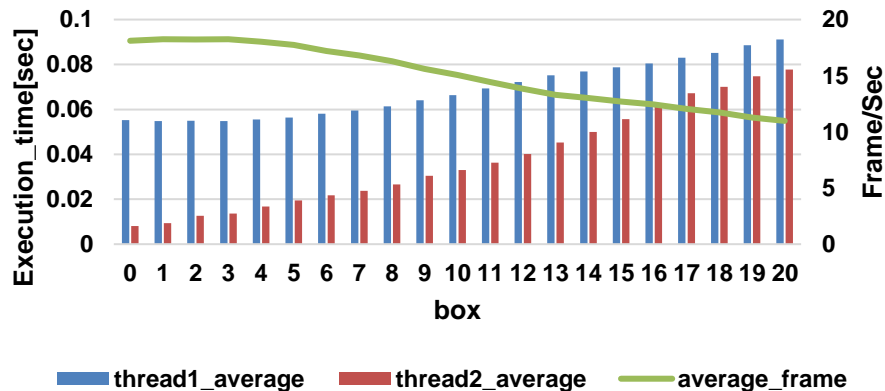


- Multithreading : 동시에 실행되는 두가지 작업 동기화

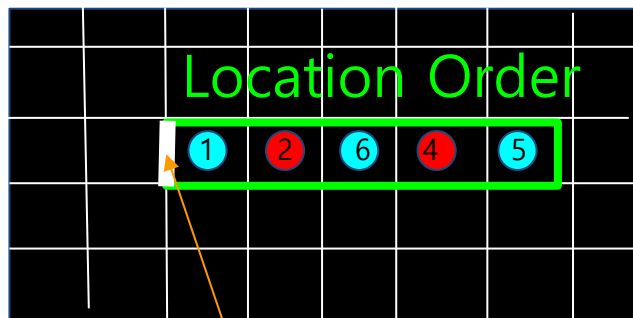
Thread Average and Frame Rate Analysis by box(Kamlan)



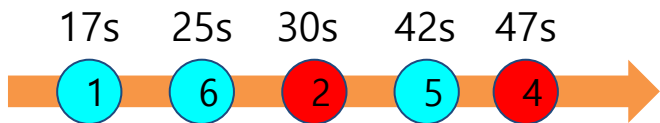
Thread Average and Frame Rate Analysis by box(IOU)



- B. 새치기 감지 알고리즘



start line



Time Order

1

```
people_in_line = {
  1: 17
  6: 25
  2: 30
  5: 42
  4: 47
}
```

ID : 들어온 시간

2

```
sorted_people_in_line = [
  [1, 17]
  [2, 30]
  [6, 25]
  [4, 47]
  [5, 42]
]
```

Sort ID in location order

3

```

17s < 25s -> pass, temp <= 17
30s > 25s -> cut in line(not update temp)
25s < 42s -> pass, temp <= 25
47s > 42s -> cut in line(not update temp)
42s          temp = 42

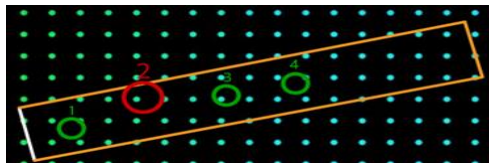
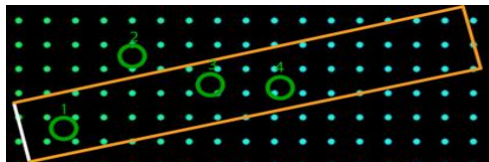
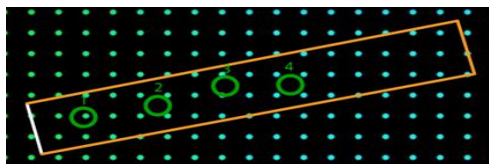
```

아래에서부터 포인터를 이동시키며 temp와 비교

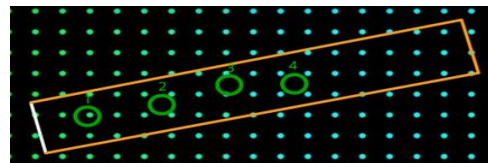
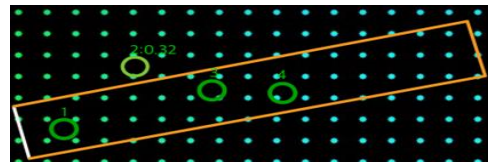
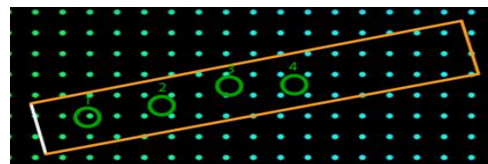
Cut in line이 아닐 때 temp update

Time Order > Location Order : 새치기로 분류

- **Corner case 1** : 줄 안의 사람이 아주 잠깐 줄 바깥으로 이동했다 들어온 경우

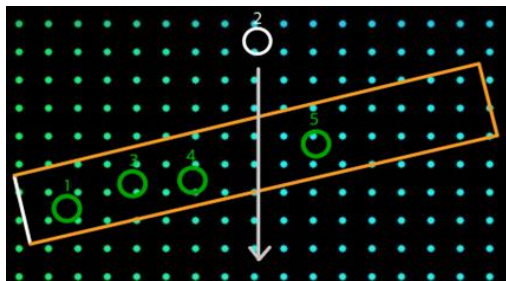


새치기로 분류

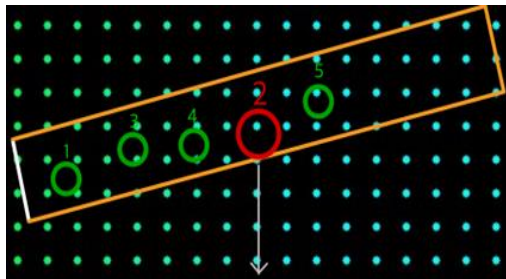


새치기로 분류되지 않음

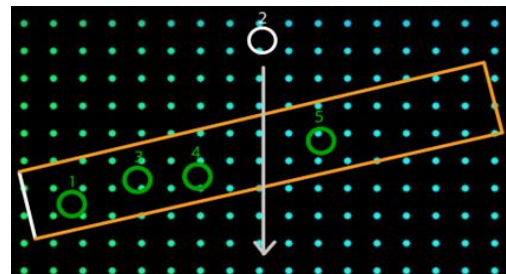
- **Corner case 2** : 줄 바깥에 있는 사람이 잠시 줄을 지나쳐 이동하거나 스치는 경우



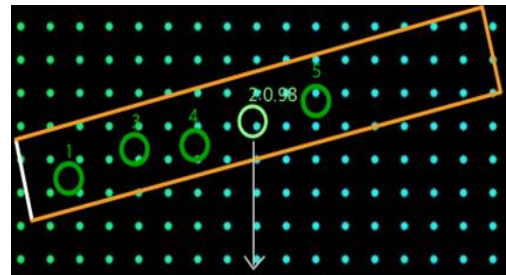
줄에 들어온 순간
people_in_line에 등록



새치기로 분류

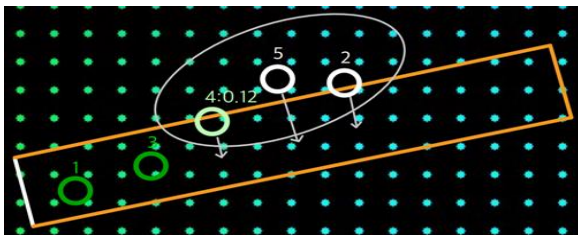


특정 시간이 지나야
people_in_line에 등록

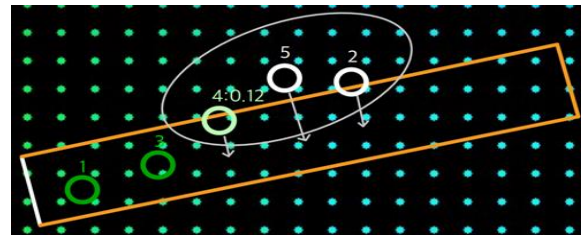
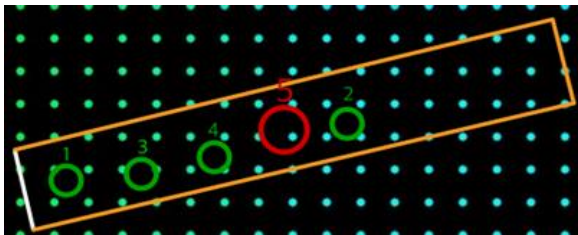


새치기로 분류되지 않음

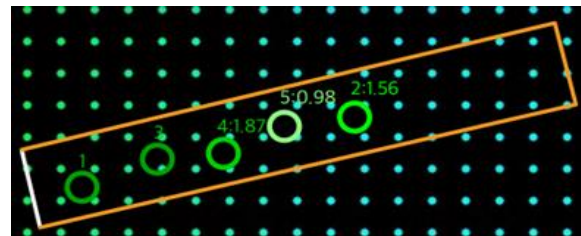
- **Corner case 3** : 들어온 시간이 비슷한 사람들(Group)끼리는 새치기를 감지하지 않도록 함



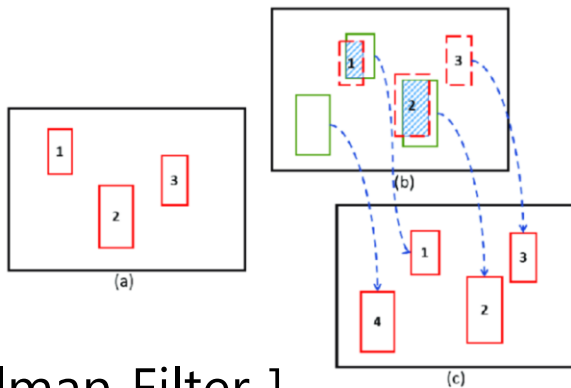
그룹으로 들어온 사람들
간 새치기 분류



들어온 시간의 차이가 **2초**
미만인 경우 새치기로 분류
하지 않음

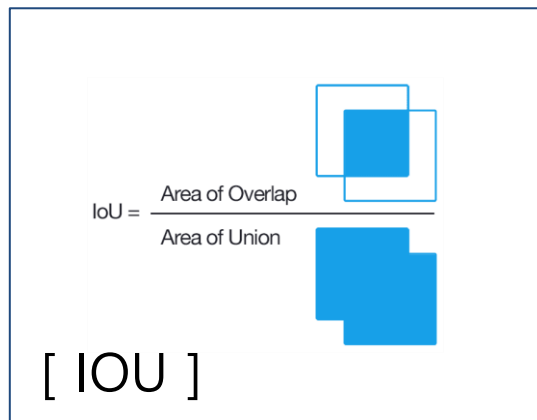


- D. Kalman Filter vs IOU



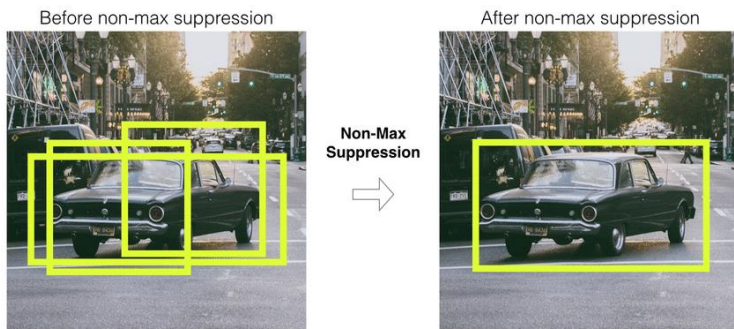
[Kalman Filter]

- Advantage: 다음 프레임 박스 위치를 예측하여 높은 추적 성능
- Disadvantage: 많은 연산량으로 인한 프레임 저하



- E1. Tracking problems by Non-Maximum Suppression(**NMS**)

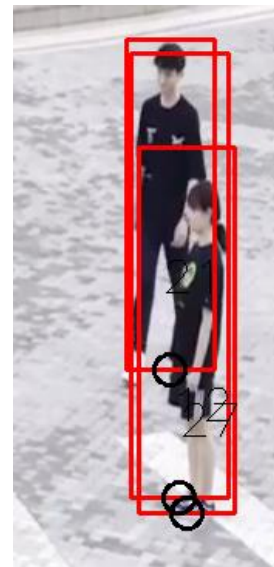
객체는 다양한 크기와 형태로 존재
-> object detection 알고리즘은 여러 개의 bounding boxes를 생성



동일한 frame에서 반환되는 한 객체에 대한
Box들이 iou_thres를 넘어가면 하나의 박스만 반환



iou_thres = 0.5



iou_thres = 0.8

- E2. Tracking problems with ID confusing

한 객체에 여러개의 box가 감지되었을 경우
ID가 바뀌어버리는 문제



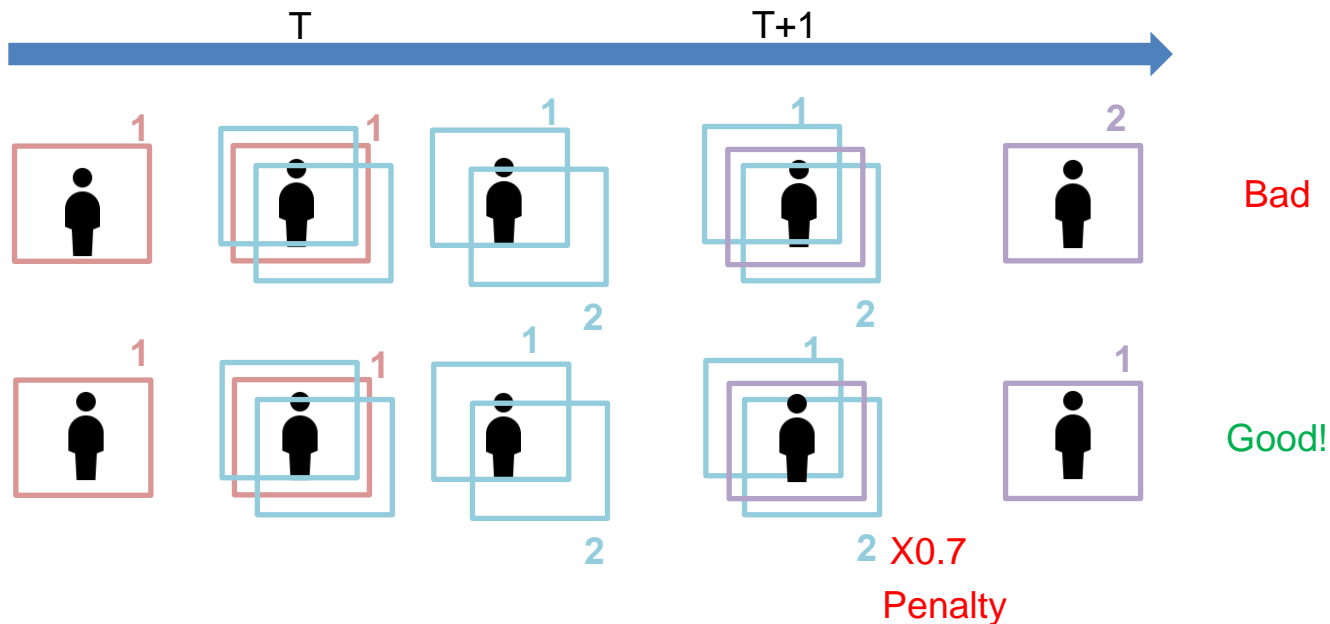
새로 생겨난 tracker는 iou를 계산할때
패널티를 줘서 기존의 ID를 대체하지 않게함



- E3. Tracking problems with ID confusing

Tracker: ID가 부여된 박스

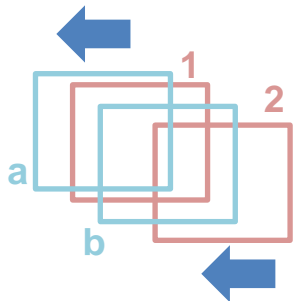
BOX: 현재 프레임에서 아직 ID가 부여되지 않은 박스
(기존ID부여 or 새로운ID부여)



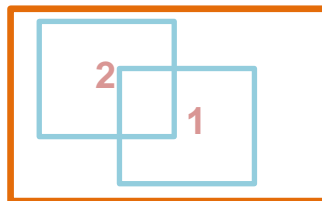
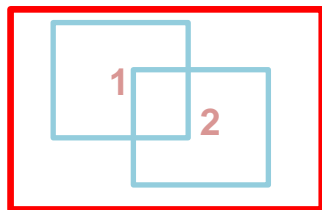
Tracker1: 기존에 존재하던 트래커

Tracker2: 15프레임 이상 지속 되지 않은 트래커 (새로운 트래커)

- E4. Tracking problems with ID confusing



IOU	a	b
1 old	0.5	0.7
2 new	0.1	0.4



Penalty	IOU_SUM	
	a-1, b-2	a-2 b-1
x0.7	$0.5 + 0.4(*0.7) = 0.78$	$0.1(*0.7) + 0.7 = 0.77$
x0.5	$0.5 + 0.4(*0.5) = 0.70$	$0.1(*0.5) + 0.7 = 0.75$

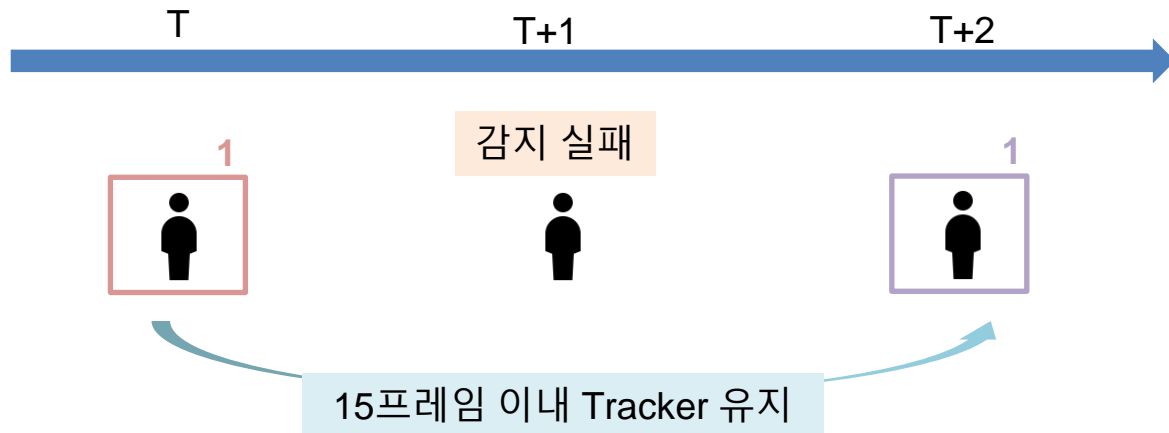
Tracker1: 기존에 존재하던 트래커

Tracker2: 15프레임 이상 지속 되지 않은 트래커 (새로운 트래커)

- E5. ID consistence

Tracker: ID가 부여된 박스

BOX: 현재 프레임에서 아직 ID가 부여되지 않은 박스
(기존ID부여 or 새로운ID부여)



```
self.people_in_line: {1: 1731252100.82, 3: 1731252105.23, 4: 1731252107.92}
sorted_people_in_line: [[4, 1731252107.92], [3, 1731252105.23], [1, 1731252100.82]]
Tracker 0: {'id': 0, 'bbox': [145.60547, 174.21875, 207.61719, 372.65625], 'unmatched_frames': 0, 'age': 512}
Tracker 1: {'id': 1, 'bbox': [482.8125, 340.8203, 538.28125, 616.40625], 'unmatched_frames': 0, 'age': 461}
Tracker 3: {'id': 3, 'bbox': [712.1094, 362.5, 770.7031, 575.78125], 'unmatched_frames': 0, 'age': 381}
Tracker 4: {'id': 4, 'bbox': [822.65625, 346.28906, 867.96875, 585.9375], 'unmatched_frames': 0, 'age': 335}
Tracker 6: {'id': 6, 'bbox': [677.7344, 284.96094, 743.3594, 514.84375], 'unmatched_frames': 0, 'age': 188}
Tracker 8: {'id': 8, 'bbox': [694.9219, 271.48438, 750.3906, 535.5469], 'unmatched_frames': 7, 'age': 11}
sorted_keys: [4, 3, 6, 1]
```

- F. Precision, Recall, mAP

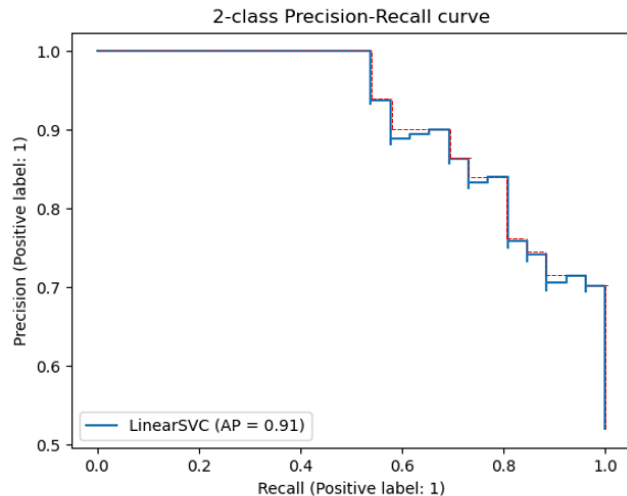
		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

$$(Precision) = \frac{TP}{TP + FP}$$

모델이 True로 분류한 것 중 진짜 True의 개수

$$(Recall) = \frac{TP}{TP + FN}$$

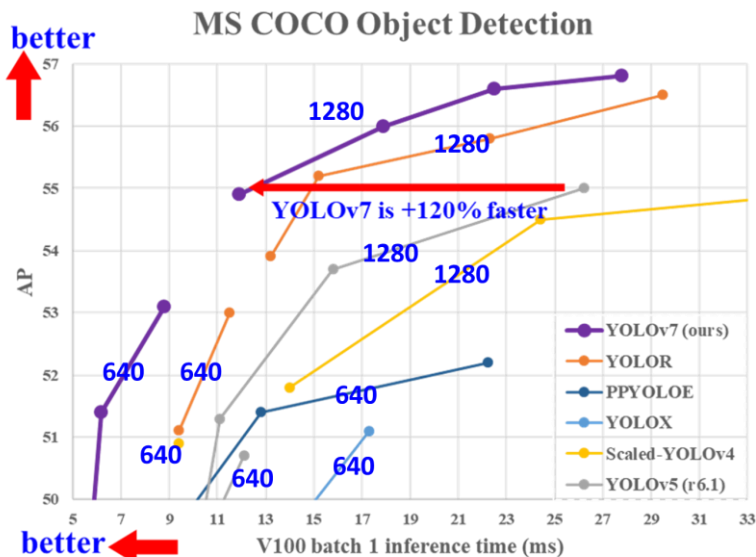
실제 True중 모델이 True로 분류한 것의 비율



AP: PR Curve의 아래 영역

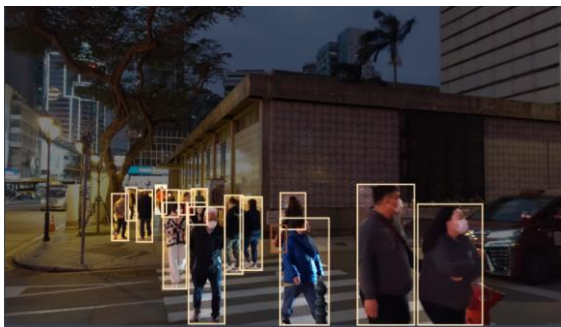
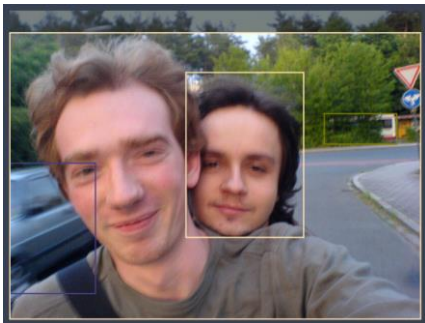
mAP: mean Average Precision, AP평균

- G. Model Select



Model	#Param.	FLOPs	Size	AP ^{val}	AP ^{val} ₅₀	AP ^{val} ₇₅	AP ^{val} _S	AP ^{val} _M	AP ^{val} _L
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X [81]	96.9M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	35.2%	52.8%	37.3%	15.7%	38.0%	53.4%
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	10.9%	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	30.8%	47.3%	32.2%	10.0%	31.9%	52.2%
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	60.4%	69.2%
YOLOv7-E6	97.2M	515.2G	1280	55.9%	73.5%	61.1%	40.6%	60.3%	70.0%
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	42.4%	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	56.8%	74.4%	62.1%	40.8%	62.1%	70.6%
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

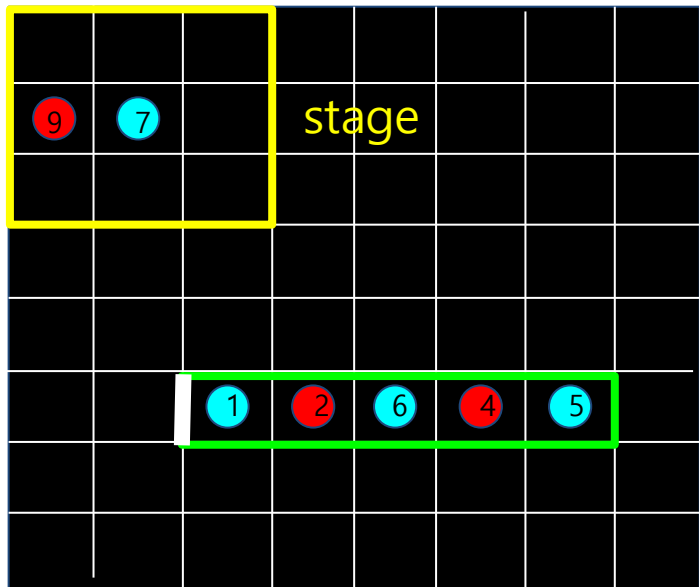
- H. People Detection Dataset



People Detection Computer Vision Project

 Leo Ueno Updated 2 months ago

- I. Photo-spot time check



Under 100s at stage

Over 100s at stage

people_in_stage = {

9: 15

7: 30

}

id: time entered in stage

if current time=120, time_limit=100

stayed_time={

9:105 -> stayed over limit

7:90 -> pass

}

- J. 기존의 프로젝트와 비교

jalotra/Queue-Detection

A naive Algorithm that uses People Detection and Convex Hull as subroutines to solve this problem:
"Given an image of..."

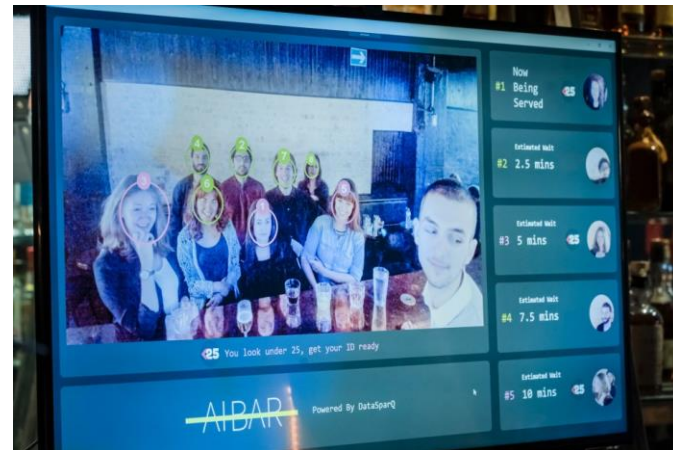
Contributor 1 Issue 1 Stars 5 Forks 0



줄에 있는 사람수 계산

예상 대기시간 계산

새치기 감지X



얼굴인식을 통해
사람별로 순서 부여

줄 감지X

- K. 다른 인공지능 플랫폼과 비교

	Jetson Nano	Jetson Xavier NX
AI performance	0.5TFLOPs	32 TFLOPs
Memory	4GB 64-bit LPDDR4	32GB 256-bit LPDDR4x
Cost	\$99	\$699



