



# DL Seminar

## DeepSORT

Simple Online and Realtime Tracking with  
a Deep Association Metric



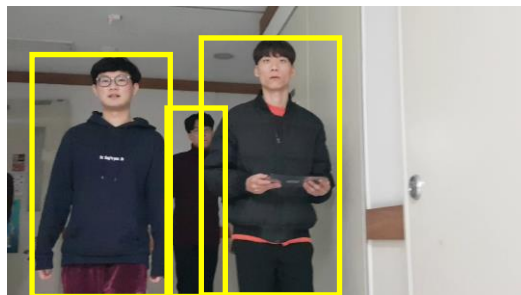
**한양대학교**  
HANYANG UNIVERSITY

인공지능 연구실  
김지성

# Introduction

## Multi Object Tracking

Frame 1



Frame 2



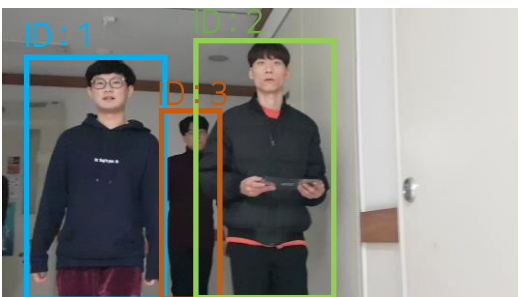
Frame N



탐지 결과

...

ID: 2



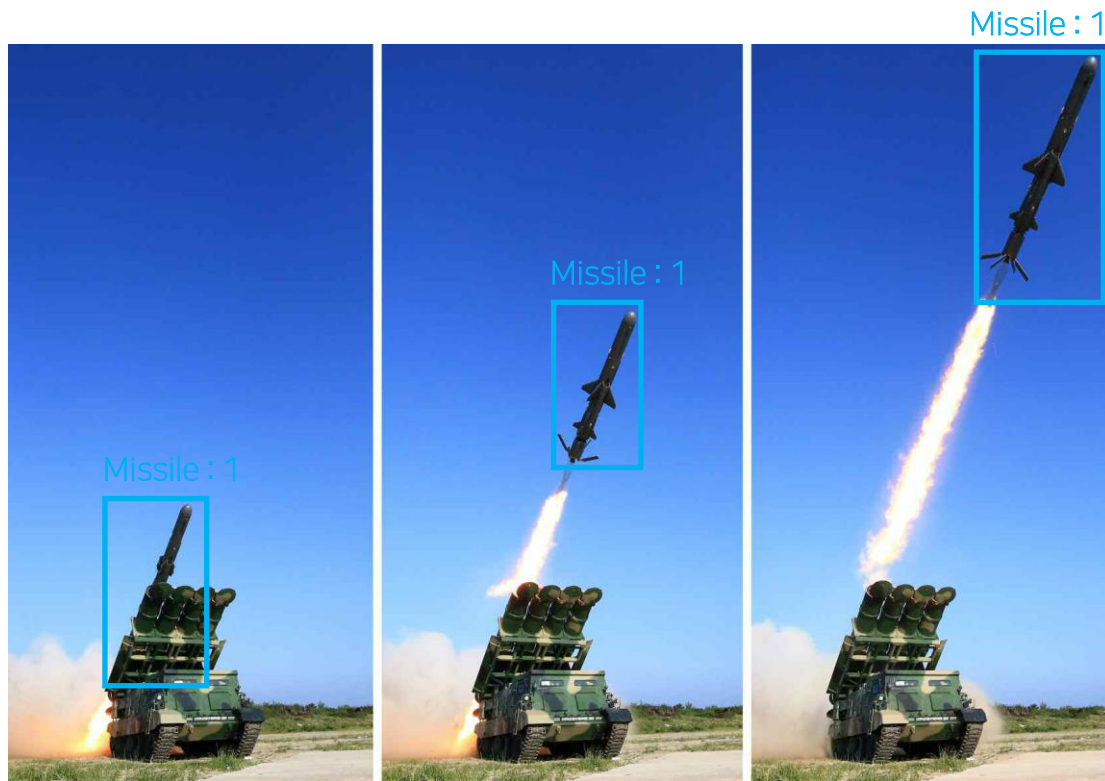
추적 결과

...

처음 탐지한 객체에 ID를 부여하고, 이후 프레임에서도 같은 ID로 인식하는 문제  
연속적인 영상(시계열 데이터)에서 시간에 따른 객체의 공간적, 시각적 변화를 얻을 수 있다.

# Introduction

## Multi Object Tracking

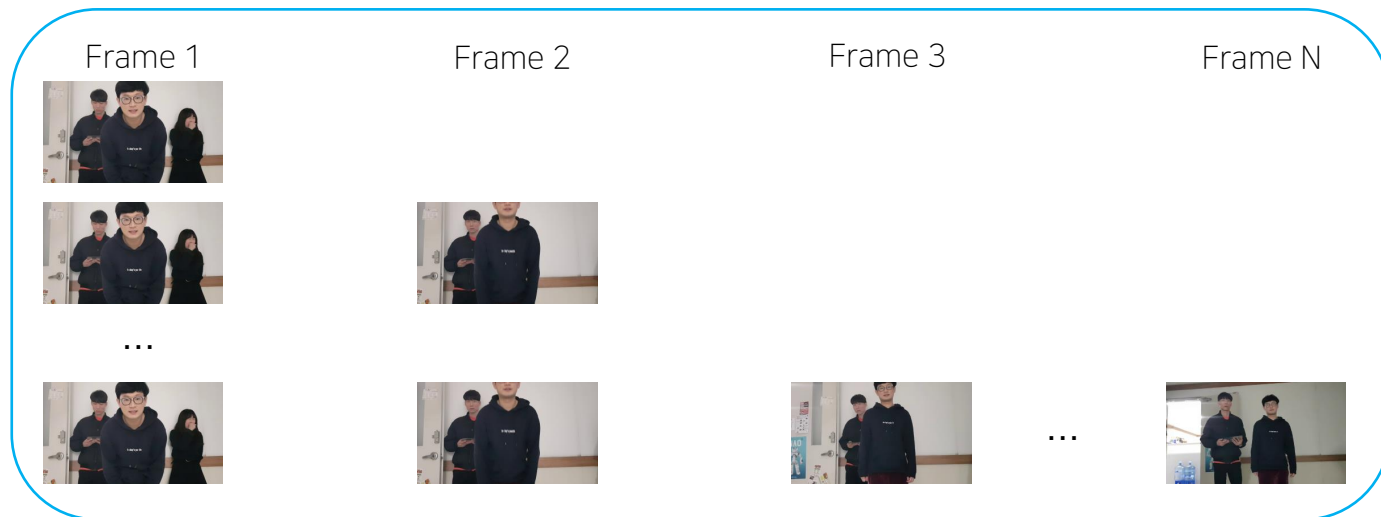


→ 다음위치는 어디?

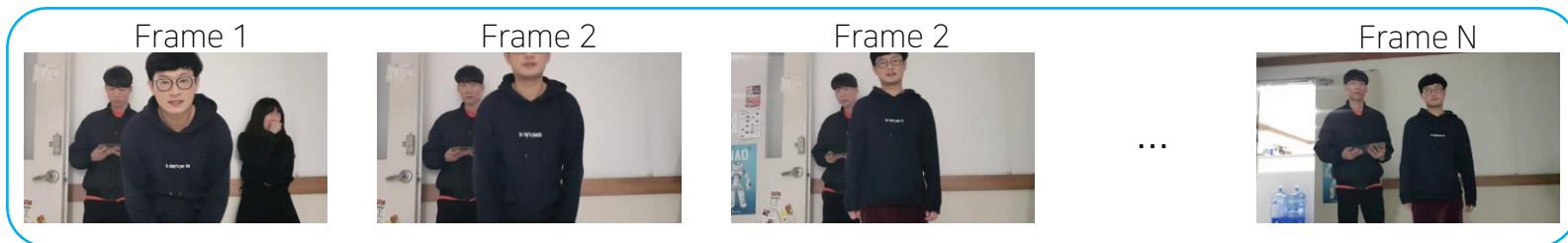
처음 탐지한 객체에 ID를 부여하고, 이후 프레임에서도 같은 ID로 인식하는 문제  
연속적인 영상(시계열 데이터)에서 시간에 따른 객체의 공간적, 시각적 변화를 얻을 수 있다.

# Introduction

## Online vs Batch



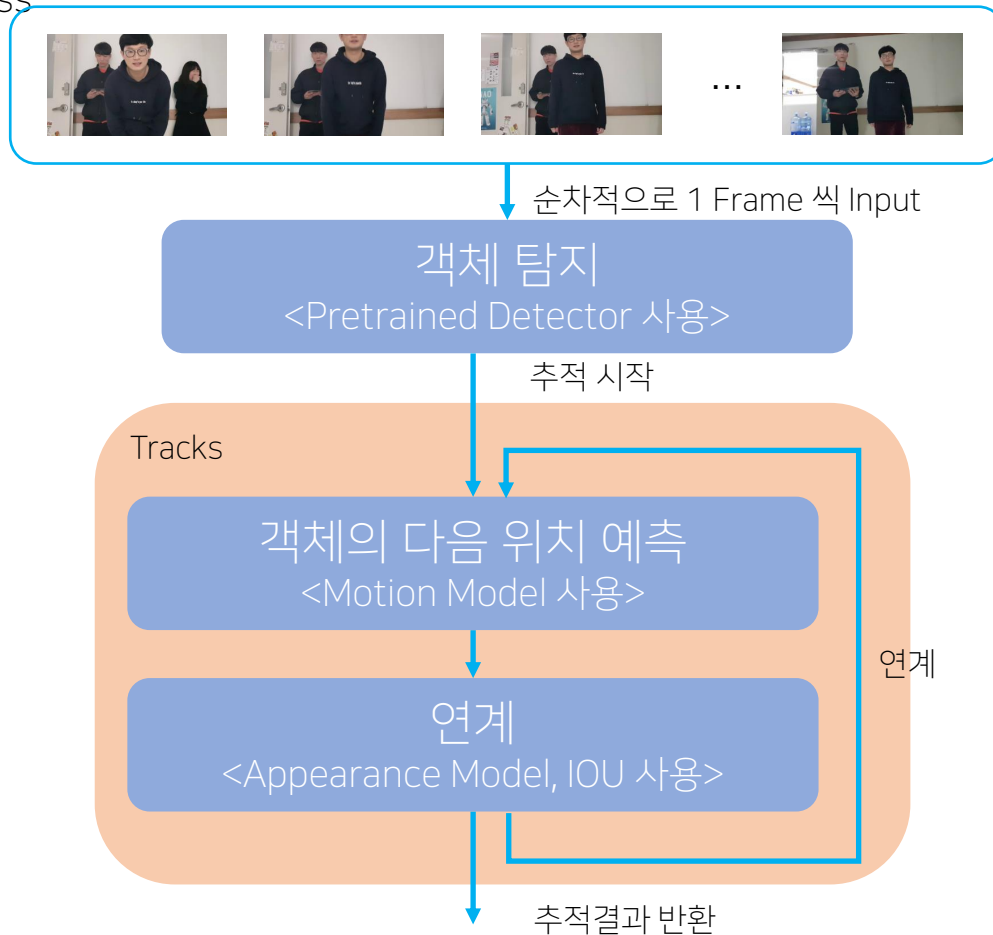
Online Tracking : 현재로부터 과거 프레임 관측 -> 실시간 추적 가능, 상대적으로 낮은 정확도



Batch Tracking : 전체 프레임 관측 -> 실시간 추적 불가능, 상대적으로 높은 정확도

# Introduction

## General Tracking Process



# Introduction

## Detect – Pretrained detector

	backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<i>Two-stage methods</i>							
Faster R-CNN+++ [3]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [6]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [4]	Inception-ResNet-v2 [19]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [18]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	<b>52.1</b>
<i>One-stage methods</i>							
YOLOv2 [13]	DarkNet-19 [13]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [9, 2]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [2]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [7]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [7]	ResNeXt-101-FPN	<b>40.8</b>	<b>61.1</b>	<b>44.1</b>	<b>24.1</b>	<b>44.2</b>	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

COCO for YOLOv3

# Introduction

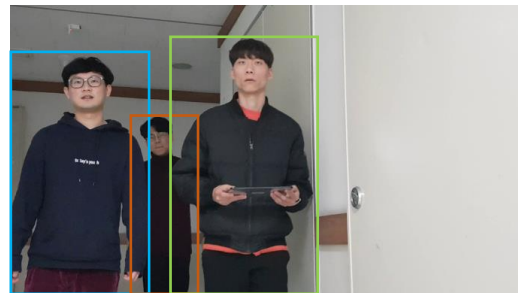
## Predict - Motion model

Frame 1

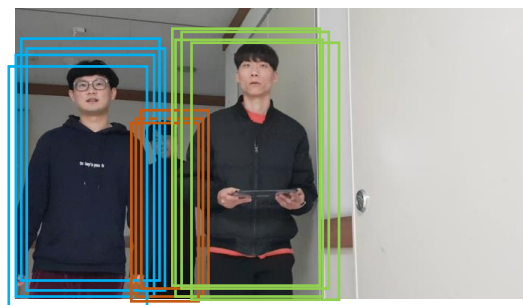
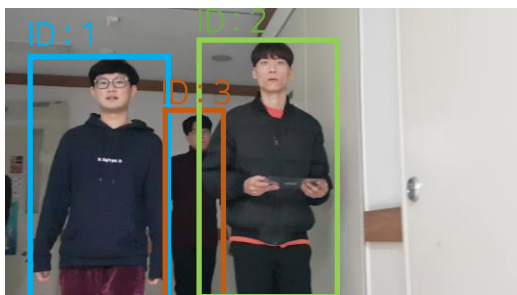


→  
칼만필터 예측

Frame 2



→  
파티클필터 예측



# Introduction

## Association

Frame 1



Frame 2



이전 위치

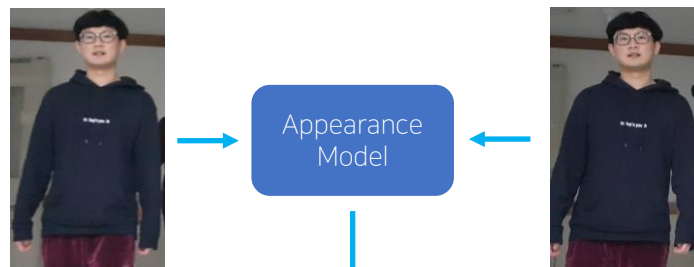
예측한 위치

IOU를 이용한 동일객체 연계 : Detector 성능에 매우 의존적



겹침 정도(IOU)

Appearance Model을 이용한 동일객체 연계 : Detect Noise 보완



객체간 유사도

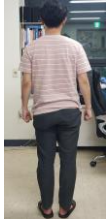


# Method

## Body Embedding



A인물의 사진1



A인물의 사진2



B인물의 사진1



Body  
Embedding



Body  
Embedding



Body  
Embedding



128d 벡터

Euclidean distance

```
array([-0.07440512, 0.13833548, 0.01550988, -0.04143589, -0.11708137,
        0.01741652, 0.09219746, -0.0411015, 0.12901564, -0.03510666,
        0.26017255, -0.01175268, -0.23214489, -0.10993981, -0.06433399,
        0.14533579, -0.18374404, -0.0706907, 0.01960303, 0.03927957,
        0.17917837, 0.10840175, 0.06728961, 0.01386871, -0.0921066,
        -0.32157087, -0.06875613, -0.11070353, 0.02105946, -0.09806988,
        0.09232023, -0.01714757, -0.16107245, -0.04865564, 0.05062422,
        0.04144309, -0.03346116, -0.03011878, 0.15263124, 0.01069992,
        -0.23673587, 0.05740198, 0.03050802, 0.2846291, 0.20607698,
        0.01160336, 0.00998583, -0.15661725, 0.09741502, -0.11773828,
        0.08130169, 0.15210505, 0.14222656, 0.02321066, 0.00600557,
        -0.07693202, -0.02959017, 0.15295519, -0.13042481, 0.03232573,
        0.1088061, -0.05199346, -0.01501178, -0.08011279, 0.17588341,
        0.02202863, 0.12124902, -0.26303217, 0.06608438, -0.123976,
        -0.14779539, 0.1516934, -0.16154116, -0.1716671, -0.25228465,
        0.01856503, 0.36660314, 0.04856375, -0.18907131, 0.05604936,
        -0.05154709, -0.04399936, 0.08519959, 0.14640823, 0.00993883,
        0.02799449, 0.11102622, 0.01848426, 0.23238975, -0.11351117,
        -0.04641433, 0.22538319, -0.00492372, 0.10828383, 0.02823116,
        0.02502055, 0.02946196, 0.0702077, -0.09549975, -0.034047,
        0.00996118, -0.08729131, -0.04567208, 0.09973253, -0.14260028,
        0.10422572, -0.00379006, 0.05201333, 0.02785442, -0.09933321,
        -0.09984644, -0.02418252, 0.13822252, -0.24259827, 0.2536359,
        0.12104575, 0.14091188, 0.07011457, 0.10865125, 0.04752614,
        0.02150964, -0.04581762, -0.23496597, -0.01059525, 0.11252803,
        -0.05433004, 0.10194337, -0.02596316])
```

```
array([-0.08902847, 0.14762324, 0.05718813, -0.05012994, -0.09676401,
        0.02028476, -0.08602992, -0.03863999, 0.10365386, -0.321321,
        0.2729257, -0.02356607, -0.23816103, -0.09732635, -0.03333118,
        0.1337371, -0.19538365, -0.07668579, 0.00275577, 0.03931012,
        0.15037588, 0.08741103, 0.0678171, 0.01460325, -0.0828172,
        -0.33090472, -0.06750867, -0.10570621, 0.01662678, -0.11775655,
        -0.10916033, -0.04655652, -0.1787895, -0.05166516, 0.04875493,
        0.0426253, -0.0591871, -0.04563718, 0.16380328, 0.00791238,
        -0.22012679, 0.04299945, 0.04911528, 0.23999902, 0.21590501,
        0.00269029, 0.00836444, -0.13134141, 0.09845343, -0.162596,
        0.11051578, 0.15520622, 0.12857951, 0.09687075, 0.01556353,
        -0.0930104, -0.04209765, 0.14811578, -0.10712323, 0.06637652,
        0.11120091, -0.05348029, -0.02427355, -0.06907345, 0.17457779,
        0.04438576, -0.13614309, -0.27176958, 0.04076207, -0.10192671,
        -0.13134097, 0.16820309, -0.17274556, -0.1703801, -0.25191918,
        0.0466072, 0.38125145, 0.03306871, -0.19173753, 0.01305585,
        -0.07093189, -0.05639979, 0.0709941, 0.15101001, 0.02203412,
        0.0188837, -0.10916664, 0.0469536, 0.2327467, -0.10933174,
        -0.06033619, 0.2151441, -0.01290991, 0.10794014, 0.02028765,
        0.05325644, -0.03414371, 0.095746, -0.1088978, -0.0816046,
        0.01456824, -0.07452301, -0.06755616, 0.11638117, -0.13528842,
        0.01066828, 0.00977207, 0.02836292, 0.02680941, -0.05801094,
        0.10640397, -0.0940248, 0.12043178, -0.23709993, 0.24780291,
        0.14884672, 0.13665006, 0.05027201, 0.09039105, 0.03249723,
        0.04752003, -0.02310374, -0.21511188, -0.00798578, 0.091371771,
        -0.01307906, 0.06952387, -0.03653527])
```

```
array([-0.07440512, 0.13833548, 0.01550988, -0.04143589, -0.11708137,
        0.01741652, 0.09219746, -0.0411015, 0.12901564, -0.03510666,
        0.26017255, -0.01175268, -0.23214489, -0.10993981, -0.06433399,
        0.14533579, -0.18374404, -0.0706907, 0.01960303, 0.03927957,
        0.17917837, 0.10840175, 0.06728961, 0.01386871, -0.0921066,
        -0.32157087, -0.06875613, -0.11070353, 0.02105946, -0.09806988,
        0.09232023, -0.01714757, -0.16107245, -0.04865564, 0.05062422,
        0.04144309, -0.03346116, -0.03011878, 0.15263124, 0.01069992,
        -0.23673587, 0.05740198, 0.03050802, 0.2846291, 0.20607698,
        0.01160336, 0.00998583, -0.15661725, 0.09741502, -0.11773828,
        0.08130169, 0.15210505, 0.14222656, 0.02321066, 0.00600557,
        -0.07693202, -0.02959017, 0.15295519, -0.13042481, 0.03232573,
        0.1088061, -0.05199346, -0.01501178, -0.08011279, 0.17588341,
        0.02202863, 0.12124902, -0.26303217, 0.06608438, -0.123976,
        -0.14779539, 0.1516934, -0.16154116, -0.1716671, -0.25228465,
        0.01856503, 0.36660314, 0.04856375, -0.18907131, 0.05604936,
        -0.05154709, -0.04399936, 0.08519959, 0.14640823, 0.00993883,
        0.02799449, 0.11102622, 0.01848426, 0.23238975, -0.11351117,
        -0.04641433, 0.22538319, -0.00492372, 0.10828383, 0.02823116,
        0.02502055, -0.02946196, 0.0702077, -0.09549975, -0.034047,
        0.00996118, -0.08729131, -0.04567208, 0.09973253, -0.14260028,
        0.10422572, -0.00379006, 0.05201333, 0.02785442, -0.09933321,
        -0.09984644, -0.02418252, 0.13822252, -0.24259827, 0.2536359,
        0.12104575, 0.14091188, 0.07011457, 0.10865125, 0.04752614,
        0.02150964, -0.04581762, -0.23496597, -0.01059525, 0.11252803,
        -0.05433004, 0.10194337, -0.02596316])
```

0.78

1.33

1.27

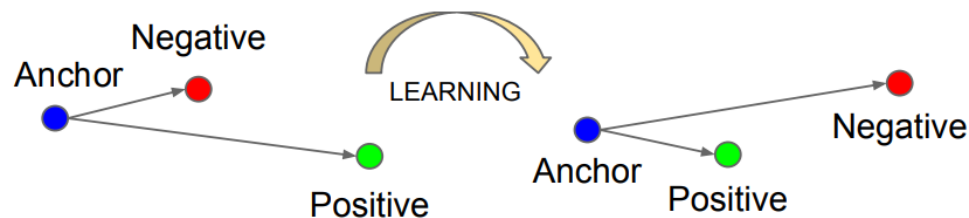
# Method

## Body Embedding

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and $\ell_2$ normalization		128

Triplet : 세 개의 데이터

- Anchor( $x_i^a$ ) : 기준 인물의 벡터
- Positive( $x_i^p$ ) : 기준과 같은 인물의 벡터
- Negative( $x_i^n$ ) : 기준과 다른 인물의 벡터



기준 인물과 같은 인물은 가깝도록, 기준 인물과 다른 인물은 멀도록

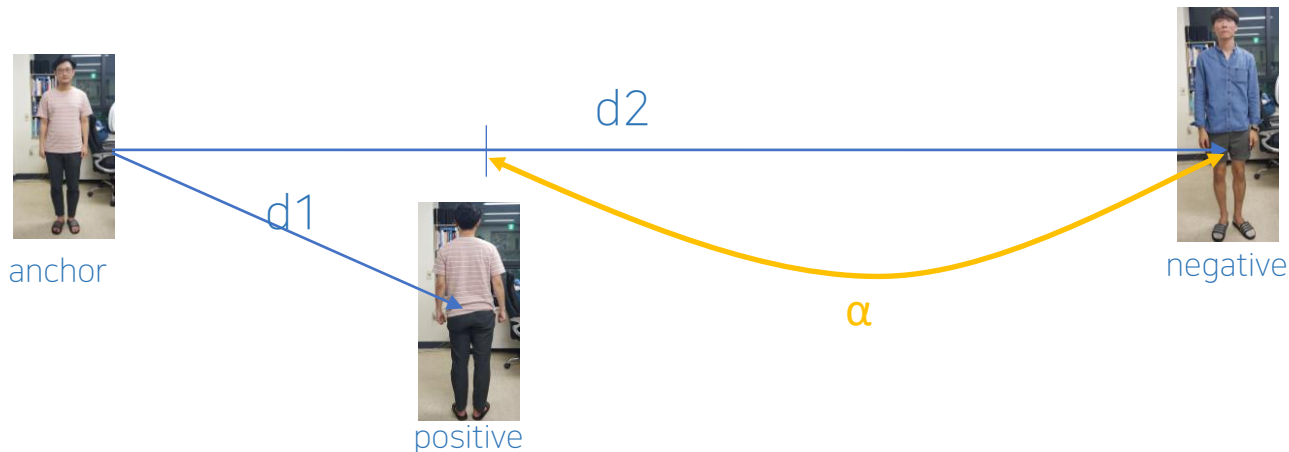
# Method

## Triplet Loss

$$\overset{d1}{\|f(x_i^a) - f(x_i^p)\|_2^2} + \alpha < \overset{d2}{\|f(x_i^a) - f(x_i^n)\|_2^2},$$

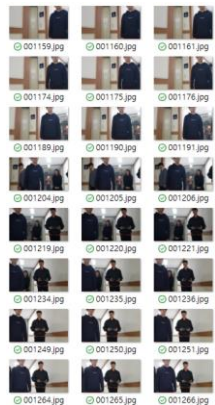
- $x$  : 이미지
- $f(x)$  : 임베딩 함수
- $\alpha$  : 마진
- $x_i^a$  : 기준 인물(anchor) 이미지
- $x_i^p$  : 기준과 같은 인물(positive)의 이미지
- $x_i^n$  : 기준과 다른 인물(negative)의 이미지

Anchor와 Negative의 제공거리가 Anchor와 Positive의 제공거리보다  $\alpha$  만큼 떨어져 있고 싶다!



# Method

## Triplet - mini batch



이미지셋

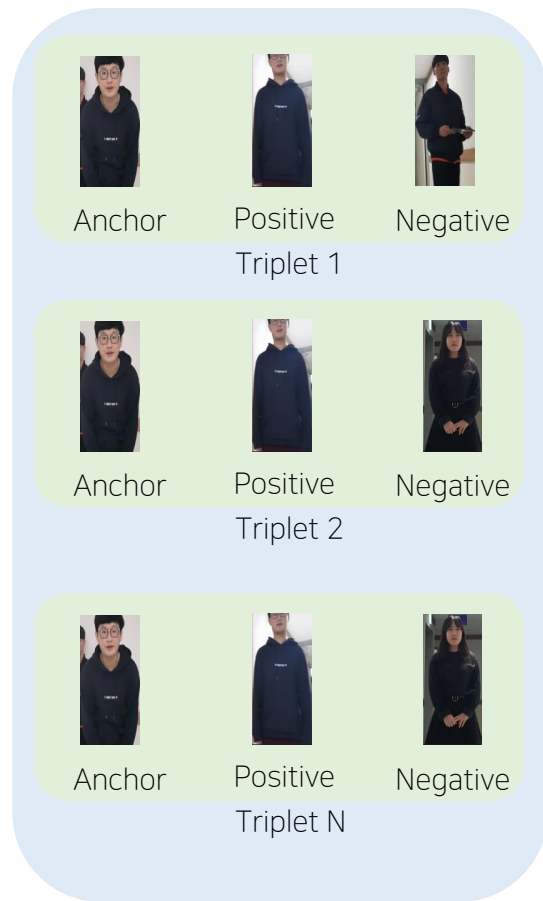
한 ID  
40개 랜덤 샘플링

Negative faces  
1960개 랜덤 샘플링



Random Sampling Batch

조합



Train Target

# Method

Triplet – mini batch

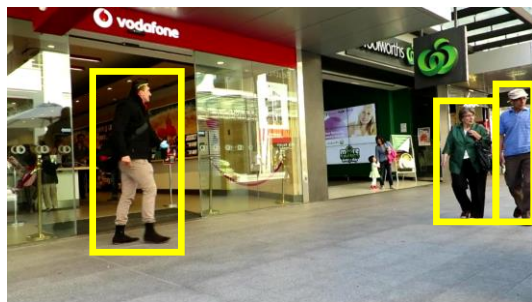


# Method

## Start Track

### 1. 탐지 결과를 사용해 트래킹 시작

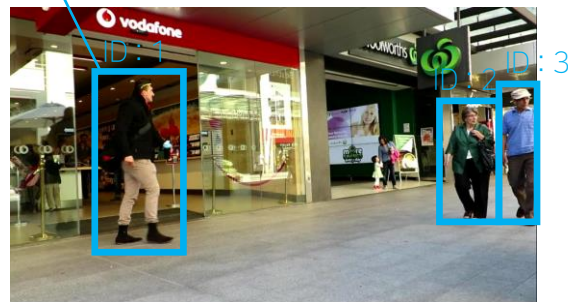
Frame 1



→  
트래킹 시작

새 트랙 생성  
`tracks += Track(id, feature)`

Frame 1

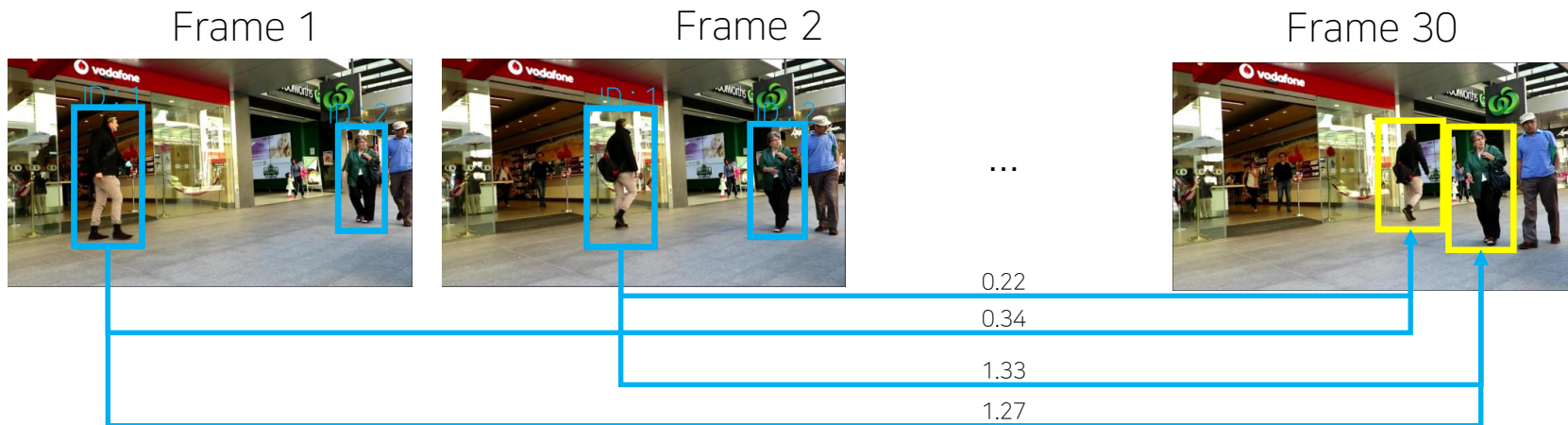


- 살아있는 트래킹
- 후보(탐지결과)

# Method

## Cascade Re-ID Matching

2. 후보와 이전 트래킹 결과를 Re-ID Model을 이용하여 매칭 - 순차적으로 과거 트래킹 피쳐와의 거리 계산



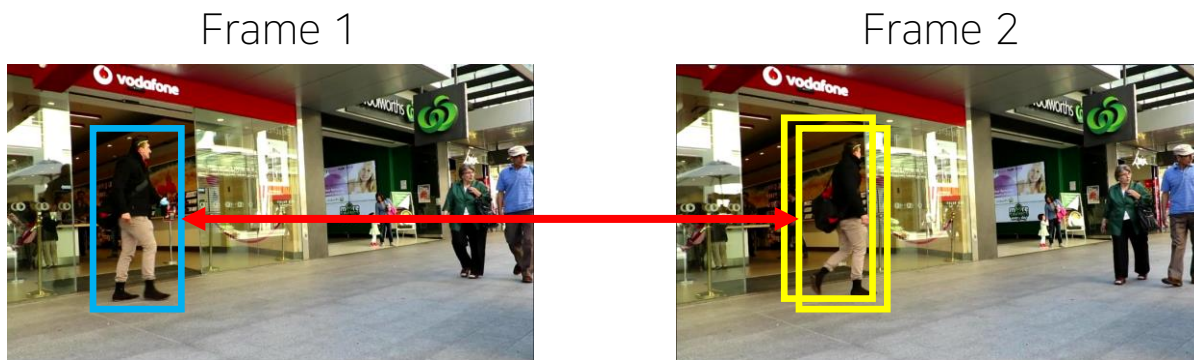
Body Feature의 L2 Distance가 Threshold 이하일 때 매칭  
(과거 피쳐 일수록 엄격한 Threshold)

- 살아있는 트래킹
- 후보(칼만필터로 예측한 위치+탐지결과)

# Method

## Association - IOU

3. 후보와 2에서 매칭되지 않은 직전 트래킹 결과를 IoU를 이용하여 매칭



칼만필터로 예측한 위치와 탐지한 위치의 IOU가 Threshold 이상일 때 매칭

□ 살아있는 트래킹

□ 후보(칼만필터로 예측한 위치+탐지결과)

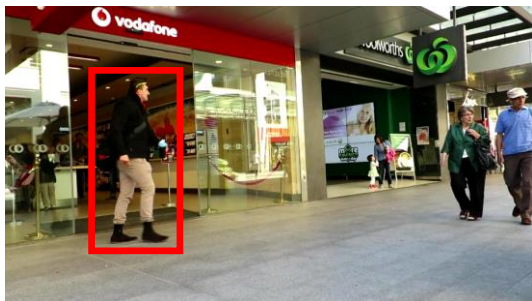


# Method

## Association - New Track

### 4. 최종적으로 매칭되지 않은 탐지결과로 새 트래킹 시작

Frame 1



Frame 2



Tracking Object를 잃어버림

Frame 3



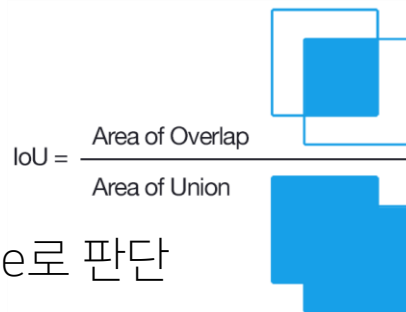
계속 Tracking Object를 잃어버림

새 트랙 생성  
`tracks += Track(id, feature)`

- ❑ 잃어버린 트래킹 결과
- ❑ 후보(탐지결과)

# Experiments

## MOT Challenge



Ground Truth와 관심영역의 IOU가 0.5 이상일 때 True로 판단

MOTA : tracker 성능지표로 적합

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t}, \quad (1)$$

t : 프레임 인덱스  
FN : 잘못탐지  
FP : 놓친 객체  
IDSW : id가 바뀐 횟수  
GT : Ground Truth의 수

MOTP : detector 성능지표로 적합

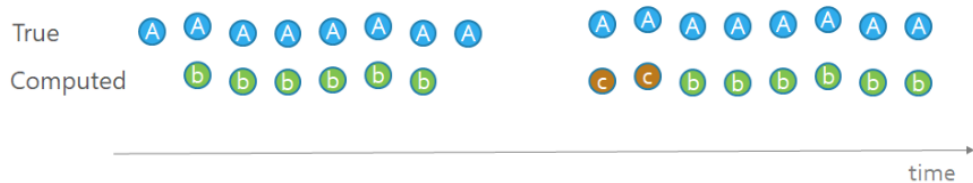
$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (2)$$

t : 프레임 인덱스  
 $c_t$  : 탐지한 객체 항목의 수  
 $d_{t,i}$  : 탐지한 객체와 Ground Truth를 비교한 IOU

# Experiments

## MOT Challenge

IDF1 : 얼마나 지속적으로 객체를 동일하게 판단하는가

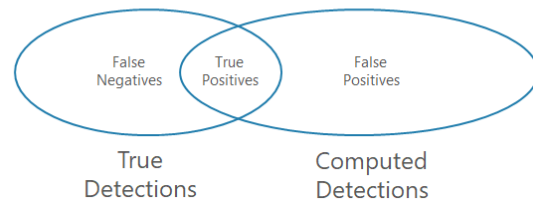


TP : 가장 많이 나온 ID의 개수 = 12

FP : 나머지 ID의 개수 = 2

FN : True - TP = 4

- ID Precision  $P = \frac{TP}{TP+FP} = \frac{TP}{C}$
- ID Recall  $R = \frac{TP}{TP+FN} = \frac{TP}{T}$
- F<sub>1</sub>-score  $F_1 = 2 \frac{PR}{P+R} = \frac{TP}{\frac{T+C}{2}}$



True Positive : True 이고, True라고 한 경우 -> 정답

False Positive : False 이지만, True라고 한 경우 -> 오답







True Negative : False 이고, False라고 한 경우 -> 정답

False Negative : True 이지만, False 라고 한 경우 -> 오답

# Experiments

## MOT Challenge

### Test Set

	Sample	Name	FPS	Resolution	Length	Tracks	Boxes	Density	Description	Source	Ref.
1		MOT16-14	25	1920x1080	750 (00:30)	164	18483	24.6	Filmed from a bus on a busy intersection	<a href="#">link</a>	[1]
2		MOT16-12	30	1920x1080	900 (00:30)	86	8295	9.2	Forward moving camera in a busy shopping mall	<a href="#">link</a>	[1]
3		MOT16-08	30	1920x1080	625 (00:21)	63	16737	26.8	A crowded pedestrian street, stationary camera	<a href="#">link</a>	[2]
4		MOT16-07	30	1920x1080	500 (00:17)	54	16322	32.6	A busy pedestrian street filmed at eye level by a moving camera	<a href="#">link</a>	[2]
5		MOT16-06	14	640x480	1194 (01:25)	221	11538	9.7	Street scene from a moving platform	<a href="#">link</a>	[3]
6		MOT16-03	30	1920x1080	1500 (00:50)	148	104556	69.7	Pedestrian street at night, elevated viewpoint	<a href="#">link</a>	[1]
7		MOT16-01	30	1920x1080	450 (00:15)	23	6395	14.2	People walking around a large square.	<a href="#">link</a>	[2]
	Total				5919 frm. (248 s.)	759	182326	30.8			

		MOTA ↑	MOTP ↑	MT ↑	ML ↓	ID ↓	FM ↓	FP ↓	FN ↓	Runtime ↑
KDNT [16]*	BATCH	68.2	79.4	41.0%	19.0%	933	1093	11479	45605	0.7 Hz
LMP_p [17]*	BATCH	<b>71.0</b>	<b>80.2</b>	<b>46.9%</b>	21.9%	434	<b>587</b>	7880	<b>44564</b>	0.5 Hz
MCMOT_HDM [18]	BATCH	62.4	78.3	31.5%	24.2%	1394	1318	9855	57257	35 Hz
NOMTwSDP16 [19]	BATCH	62.2	79.6	32.5%	31.1%	<b>406</b>	642	<b>5119</b>	63352	3 Hz
EAMTT [20]	<b>ONLINE</b>	52.5	78.8	19.0%	34.9%	910	<b>1321</b>	<b>4407</b>	81223	12 Hz
POI [16]*	<b>ONLINE</b>	<b>66.1</b>	79.5	<b>34.0%</b>	20.8%	805	3093	5061	<b>55914</b>	10 Hz
SORT [12]*	<b>ONLINE</b>	59.8	<b>79.6</b>	25.4%	22.7%	1423	1835	8698	63245	<b>60 Hz</b>
Deep SORT (Ours)*	<b>ONLINE</b>	61.4	79.1	32.8%	<b>18.2%</b>	<b>781</b>	2008	12852	56668	40 Hz



감사합니다.