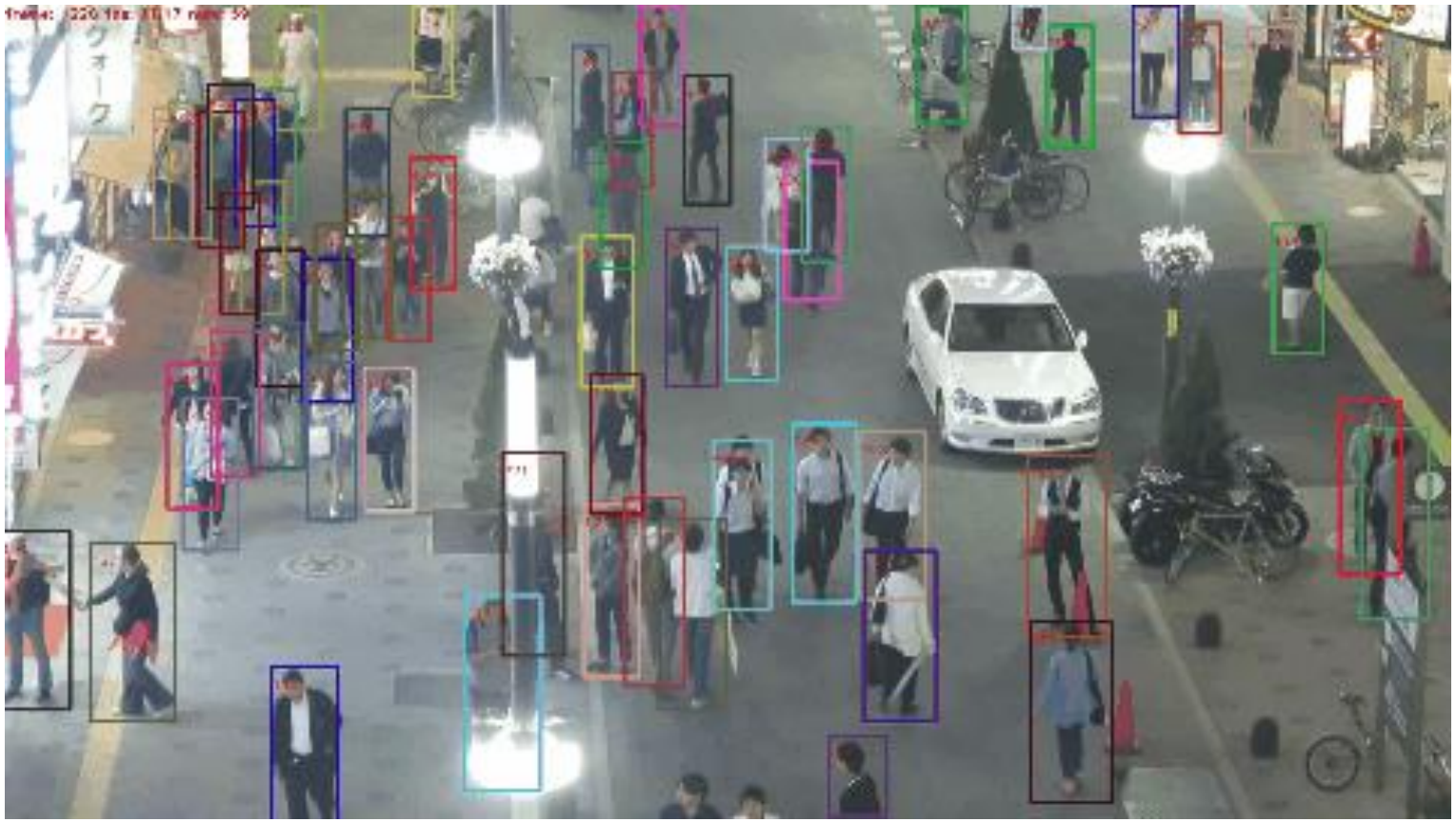
https://arxiv.org/pdf/1909.12605v1.pdf

Towards Real-Time Multi-Object Tracking

Zhongdao Wang1 Liang Zheng2 Yixuan Liu1 Shengjin Wang1
1 Department of Electronic Engineering, Tsinghua University
2 Austrilian National University
2019

인공지능 연구실
석사과정 구자봉

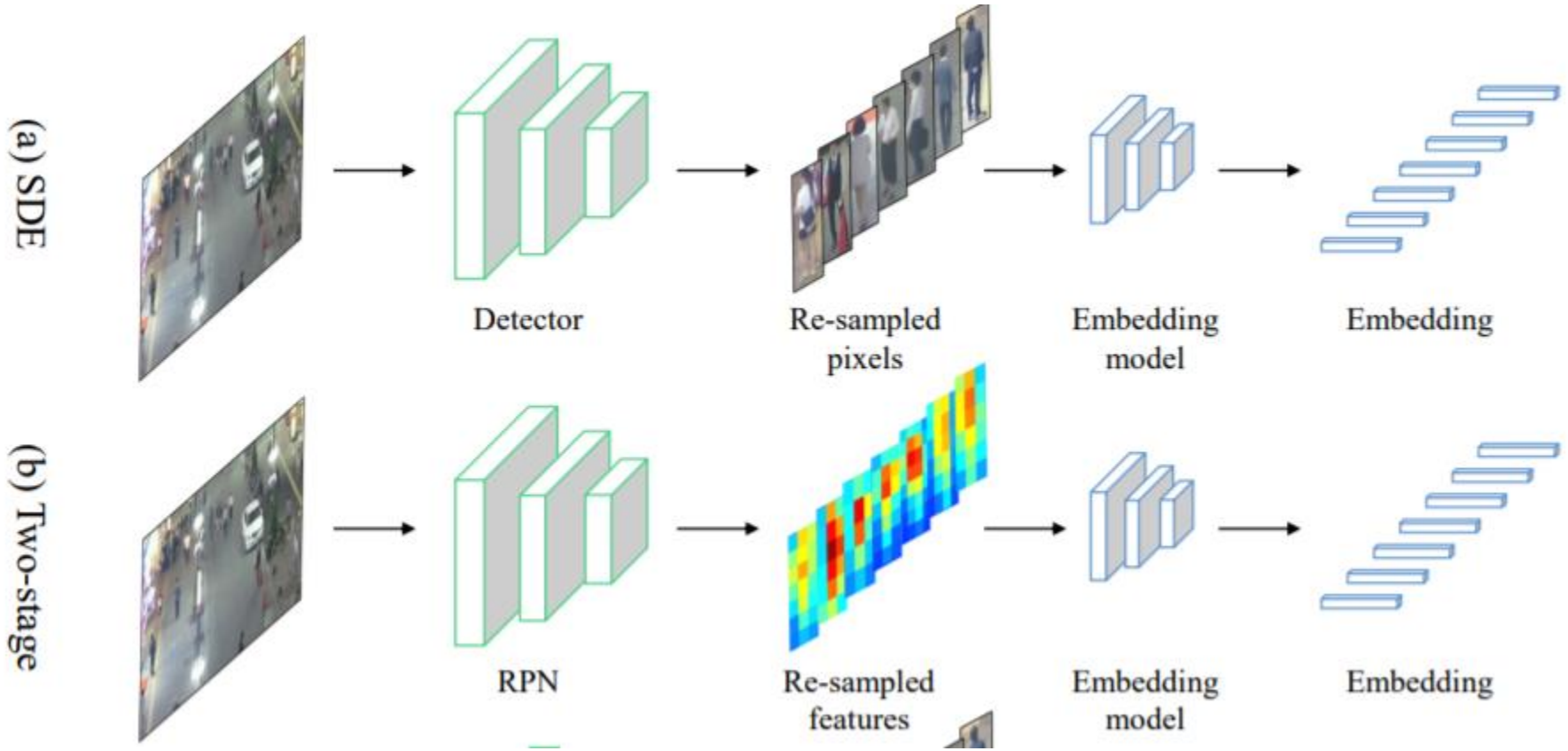
문제 정의 :
MOT(multiple object tracking)



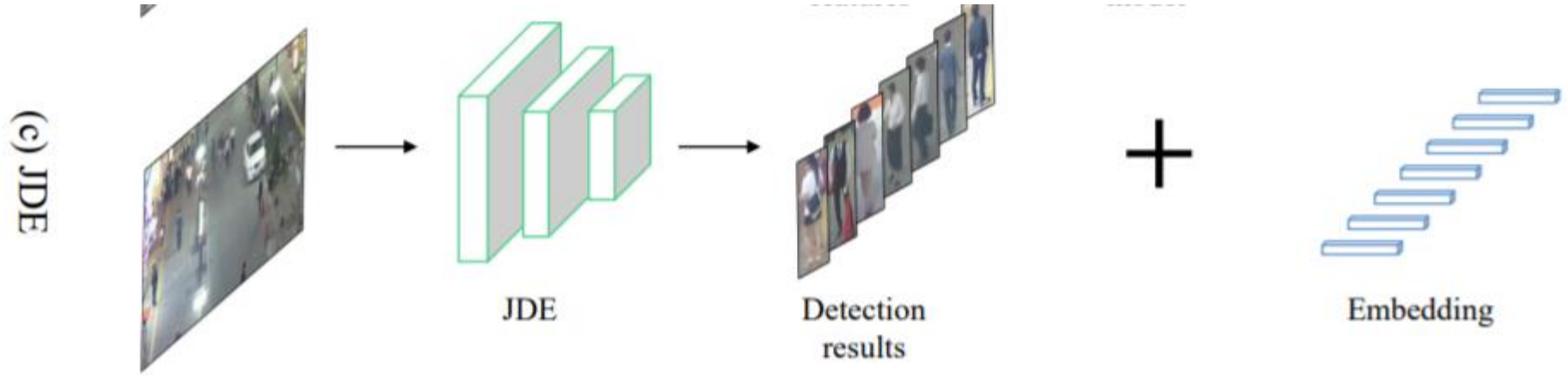
기존 방법의 문제점 제시 :

(a) 두 단계로 나뉘어서 실행됨(속도 느림)

(b) two-stage model 속도가 느림(속도 10fps)



목표 :
단일 네트워크로 디텍팅과 임베딩을 하여 속도를 높임과 동시에 성능을 만족스럽게 한다



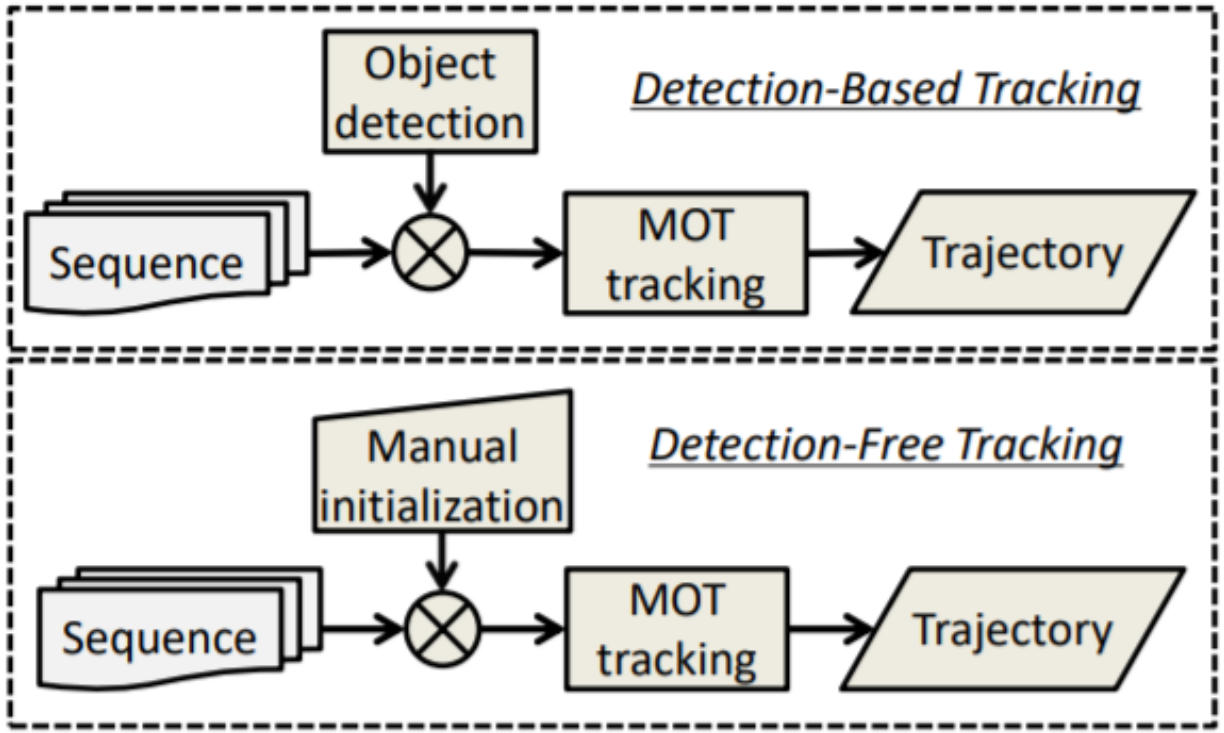
https://arxiv.org/pdf/1409.7618.pdf

다중 객체 추정 MOT (Multiple Object Tracking) :

a) initialization method
b) processing mode
c) type of output

https://arxiv.org/pdf/1409.7618.pdf

다중 객체 추정 MOT (Multiple Object Tracking) : a) initialization method (DBT, DFT)

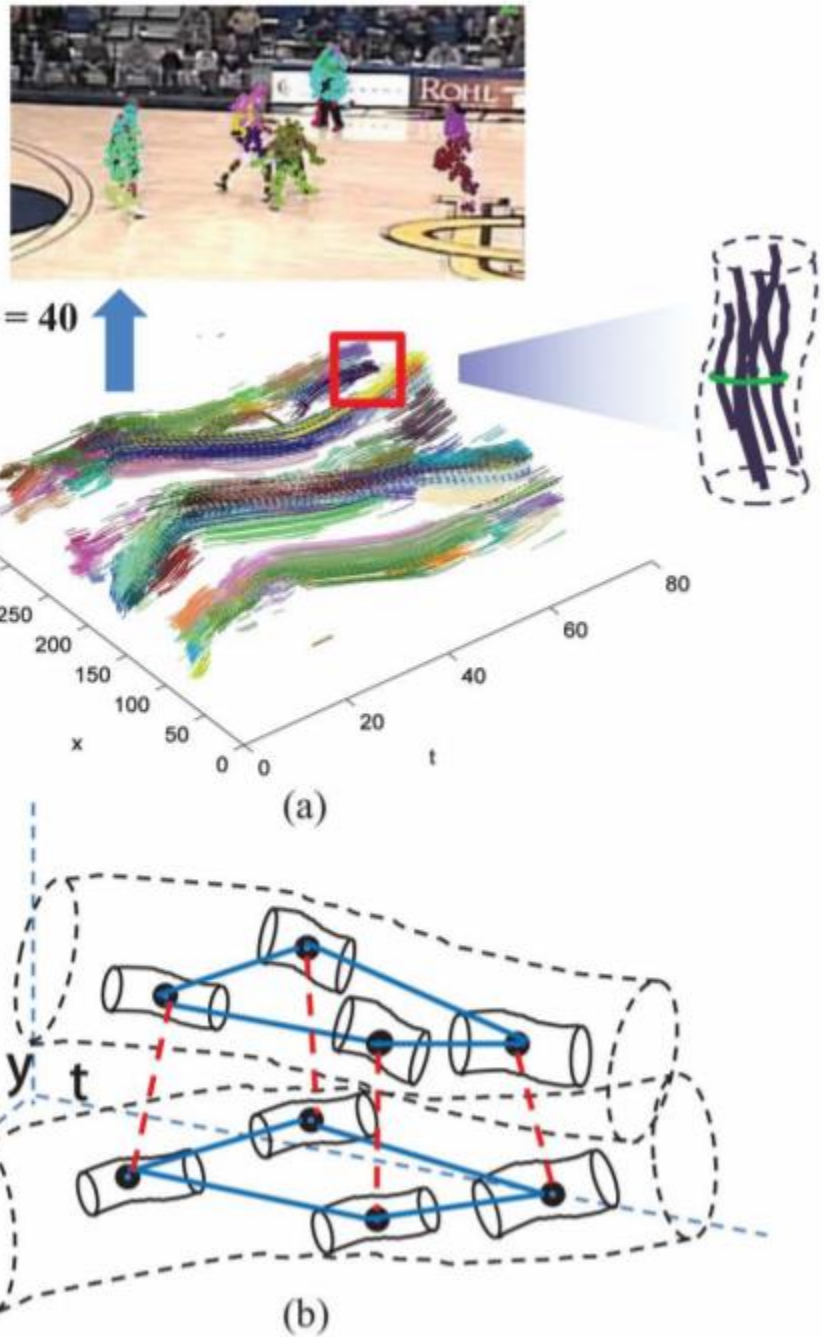


Item	DBT	DFT
Initialization	automatic, imperfect	manual, perfect
# of objects	varying	fixed
Applications	specific type of objects (in most cases)	any type of objects
Advantages	ability to handle varying number of objects	free of object detector
Drawbacks	performance depends on object detection	manual initialization

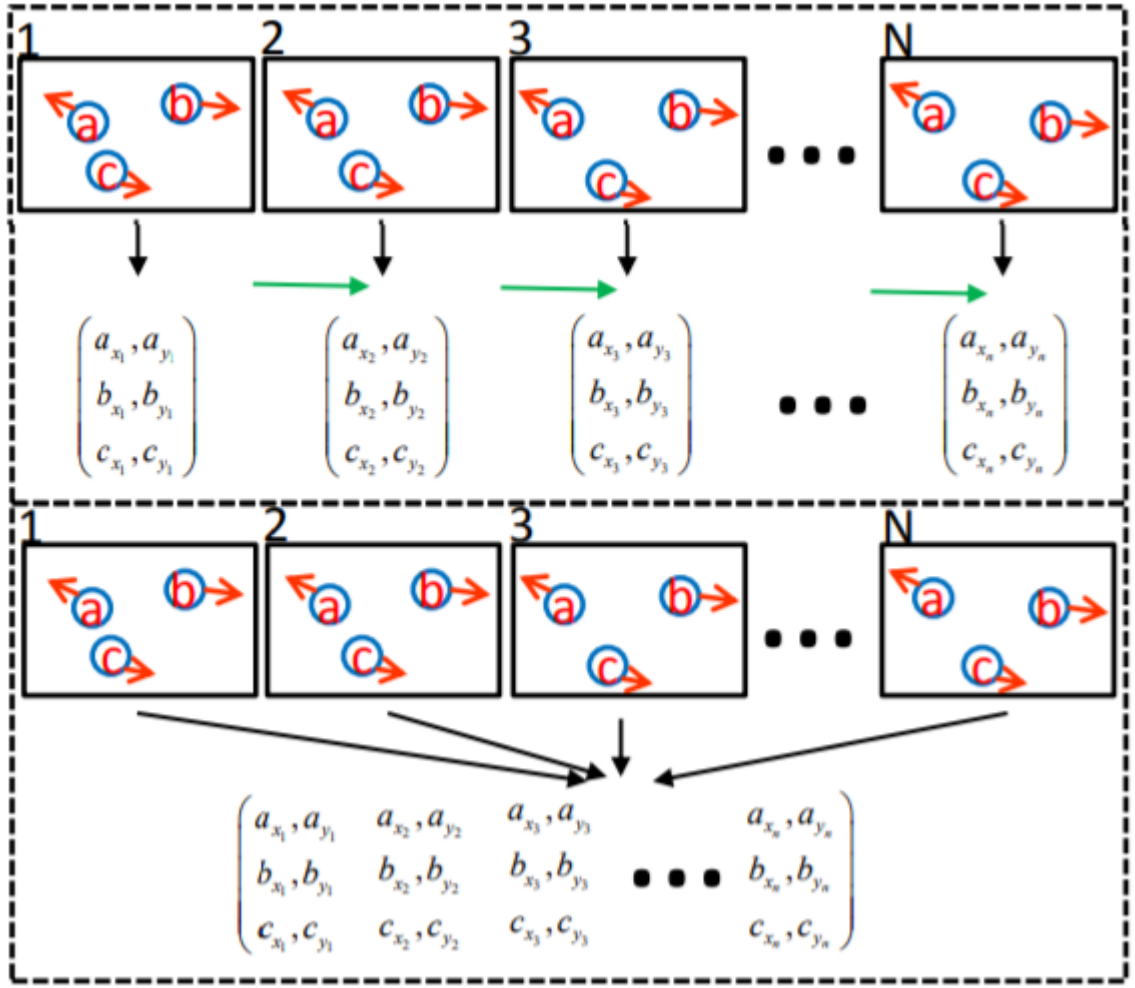
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7283579

(a)

(b)



다중 객체 추정 MOT (Multiple Object Tracking) : b) processing mode (Online, Offline)



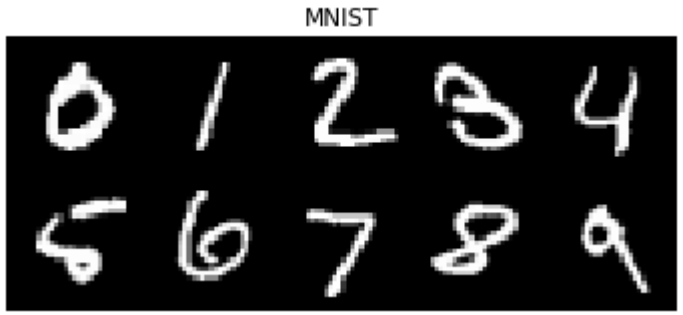
Item	Online tracking	Offline tracking
Input	Up-to-time observations	All observations
Methodology	Gradually extend existing trajectories with current observations	Link observations into trajectories
Advantages	Suitable for online tasks	Obtain global optimal solution theoretically
Drawbacks	Suffer from shortage of observation	Delay in outputting final results

다중 객체 추정 MOT (Multiple Object Tracking) : c) type of output (a에 따라 다름)

https://jsideas.net/ImageEmbeddingWithTripletLoss/ https://wwiiiii.tistory.com/entry/Pairwise-Triplet-Loss

Image Embedding :

Embedding은 고차원의 정보를 상대적으로 낮은 차원으로 변환하는 것을 의미한다. 아무 숫자로 바꾸는 것이 아니라 **정보를 보존**해야 한다.



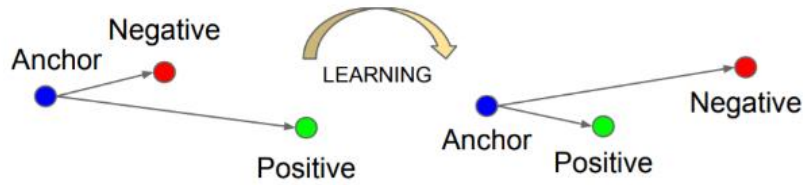
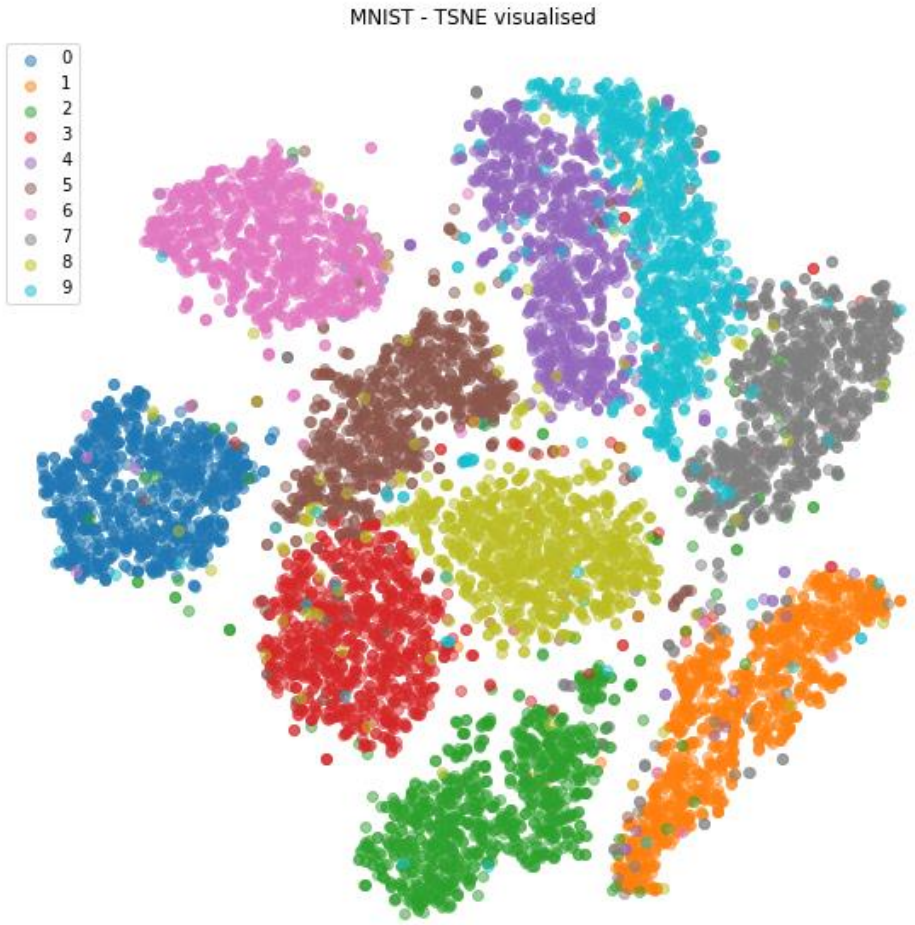


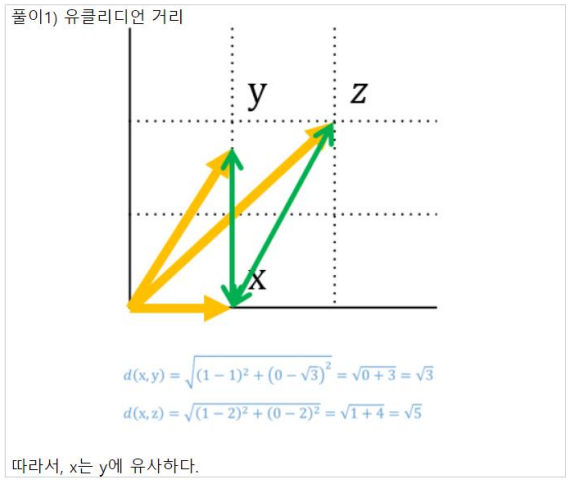
Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

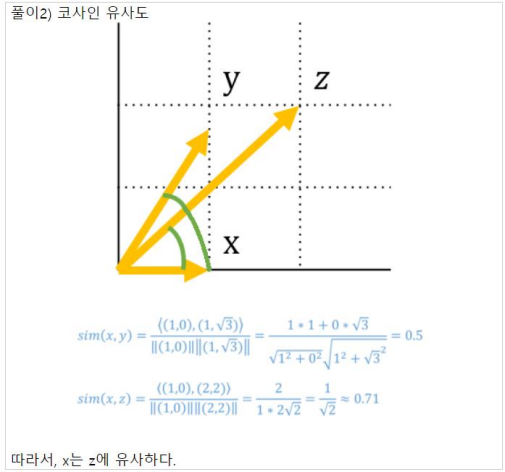


https://m.blog.naver.com/PostView.nhn?blogId=cjh226&logNo=220810613028&proxyReferer=https%3A%2F%2Fwww.google.com%2F

Image Embedding :

Embedding은 고차원의 정보를 상대적으로 낮은 차원으로 변환하는 것을 의미한다. 아무 숫자로 바꾸는 것이 아니라 **정보를 보존**해야 한다.





$$d(x,y) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2}$$

where $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n)$

$$sim(x,y) = \frac{\langle x,y \rangle}{\|x\|\|y\|}$$

where $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n)$

$$\|x\| = \sqrt{\sum_{i=1}^{n}(x_i)^2}, \langle x,y \rangle = \sum_{i=1}^{n} x_i y_i$$

제안하는 모델 :

인풋(비디오)

$\{\mathbf{I}, \mathbf{B}, \mathbf{y}\}_{i=1}^{N}$

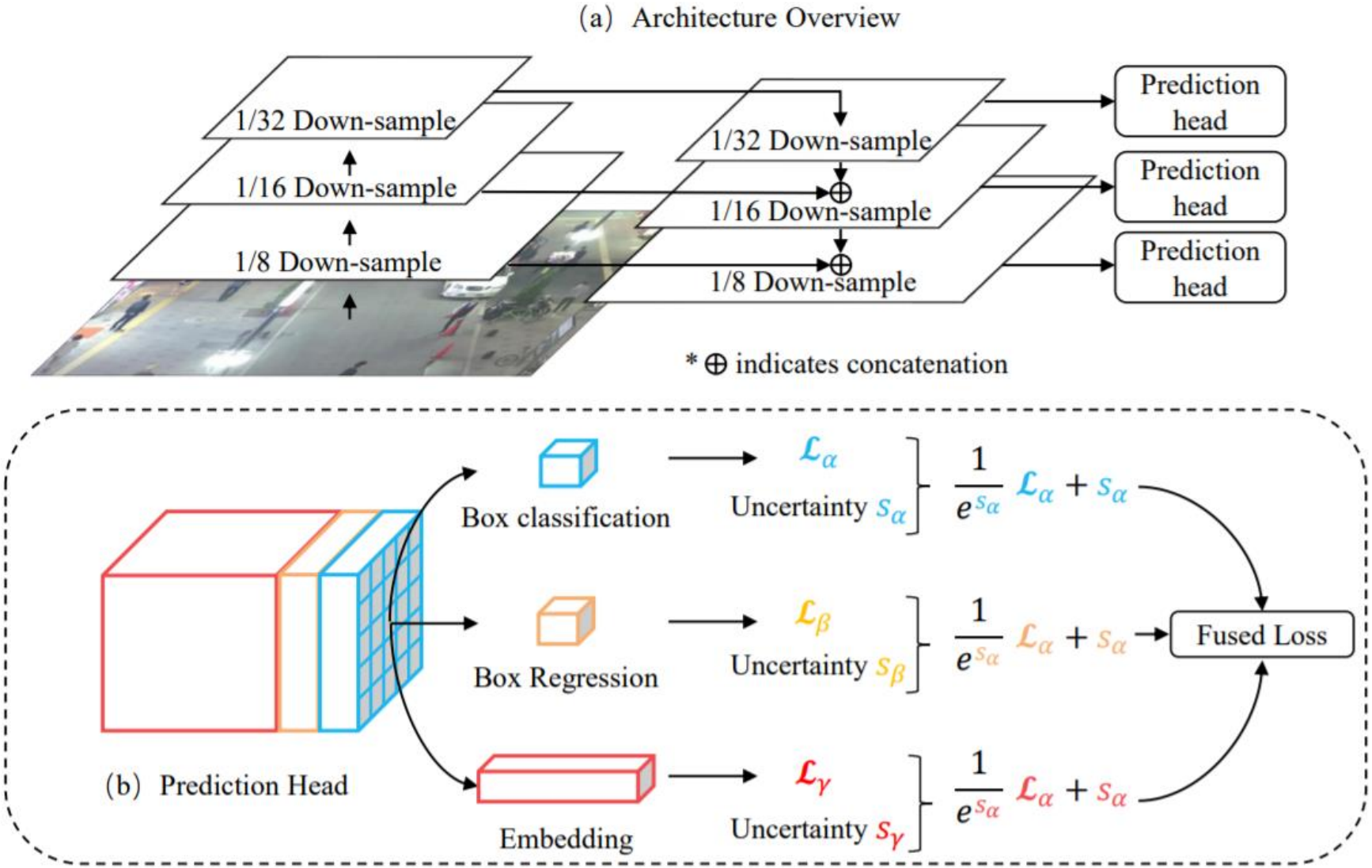
$\mathbf{I} \in \mathbb{R}^{c \times h \times w}$

$\mathbf{B} \in \mathbb{R}^{k \times 4}$

$\mathbf{y} \in \mathbb{Z}^{k}$

$\hat{\mathbf{B}} \in \mathbb{R}^{\hat{k} \times 4}$

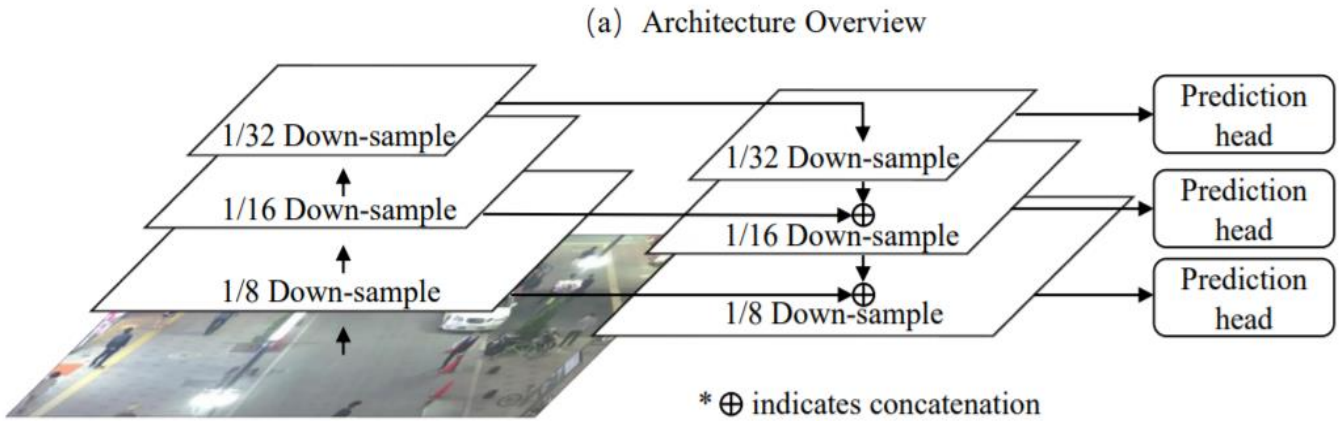
$\hat{\mathbf{F}} \in \mathbb{R}^{\hat{k} \times D}$



오브젝트 디텍팅 부분 :

기존 FPN과 다른점 :

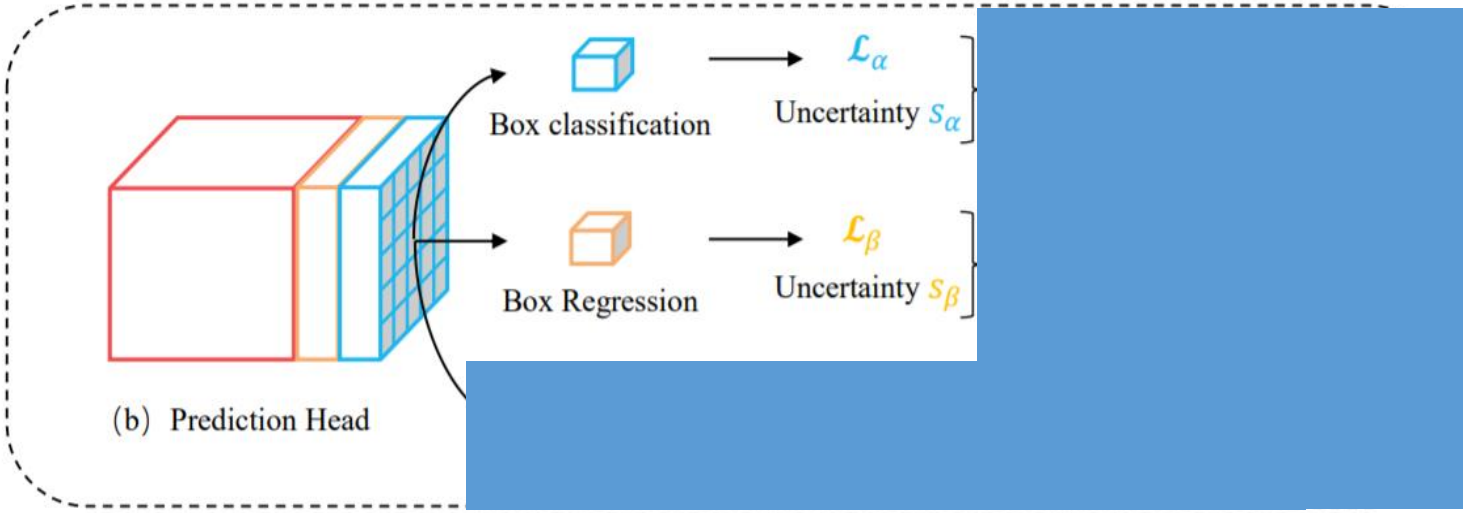
1) 인간 위주의 디텍션을 위해 앵커 사이즈 변경(1:3 종횡비)
2) IoU>0.5 이상 사용, IoU <0.4 면 배경으로



$\mathbf{B}^*$ is as close to $\mathbf{B}$ as possible

$\mathcal{L}_\alpha$ 크로스 엔트로피

$\mathcal{L}_\beta$ L1로스



임베딩 :

1) 인간 위주의 디텍션을 위해 앵커 사이즈 변경(1:3 종횡비)
2) IoU>0.5 이상 사용, IoU <0.4 면 배경으로

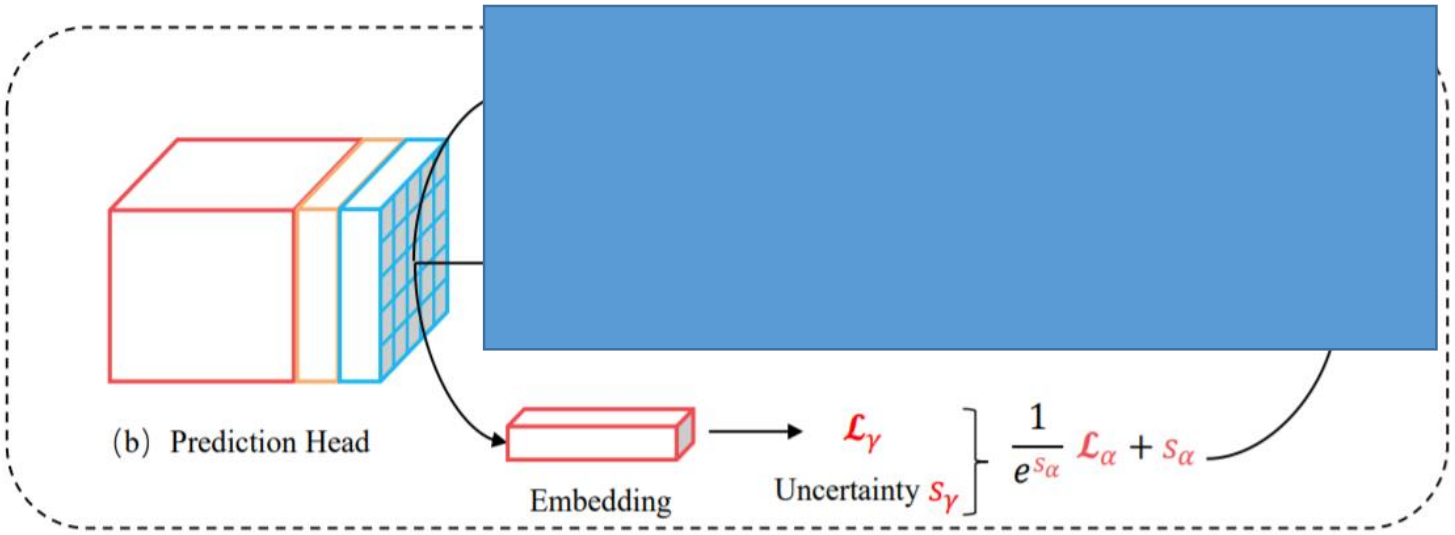
$$d(\cdot),\ \forall(k_t, k_{t+\Delta t}, k'_{t+\Delta t})$$

$$\mathbf{y}_{k_{t+\Delta t}} = \mathbf{y}_{k_t} \qquad \mathbf{y}_{k'_{t+\Delta t}} \neq \mathbf{y}_{k_t},$$

$$d(f_{k_t}, f_{k_{t+\Delta t}}) < d(f_{k_t}, f_{k'_{t+\Delta t}})$$

$$\mathcal{L}_{triplet} = \max(0, f^\top f^- - f^\top f^+)$$

$$\mathcal{L}_{CE} = -\log \frac{\exp(f^\top g^+)}{\exp(f^\top g^+) + \sum_i \exp(f^\top g_i^-)},$$



제안하는 모델 :

$$\mathcal{L}_{total} = \sum_{i}^{M} \sum_{j=\alpha,\beta,\gamma} w_j^i \mathcal{L}_j^i,$$

여러 번 랜덤하게 돌리면 잘 나옴

$$\mathcal{L}_{total} = \sum_{i}^{M} \sum_{j=\alpha,\beta,\gamma} \frac{1}{2}\left(\frac{1}{e^{s_j^i}}\mathcal{L}_j^i + s_j^i\right),$$

불확실성을 제거하기 위해 학습가능한 매개 변수를 추가함

데이터셋

Dataset	ETH	CP	CT	M16	CS	PRW	Total
# img	2K	3K	27K	53K	11K	6K	54K
# box	17K	21K	46K	112K	55K	18K	270K
# ID	-	-	0.6K	0.5K	7K	0.5K	8.7K

Table 1: Statistics of the joint training set.

https://motchallenge.net/results/3D_MOT_2015/?chl=3&orderBy=MOTA&orderStyle=DESC&det=Public

평가 지표 3가지 (디텍팅, 임베딩, 추적을 위한)

MOTA	higher	100 %	Multiple Object Tracking Accuracy [1]. This measure combines three error sources: false positives, missed targets and identity switches.

$$\mathrm{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

$$\overline{m} = \frac{\sum_t m_t}{\sum_t g_t} \quad \overline{fp} = \frac{\sum_t fp_t}{\sum_t g_t} \quad \overline{mme} = \frac{\sum_t mme_t}{\sum_t g_t}$$

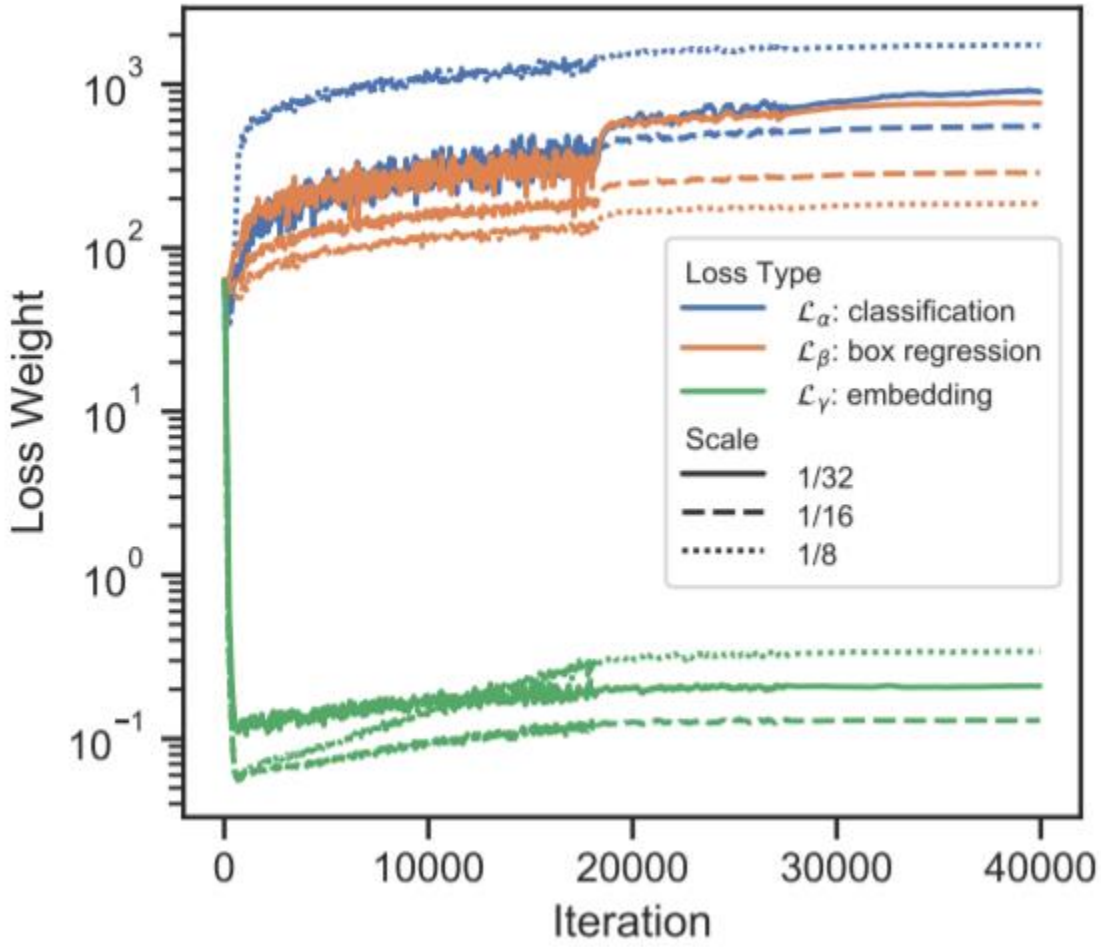
number of misses, of false positives, and of mismatches, respectively, for time t

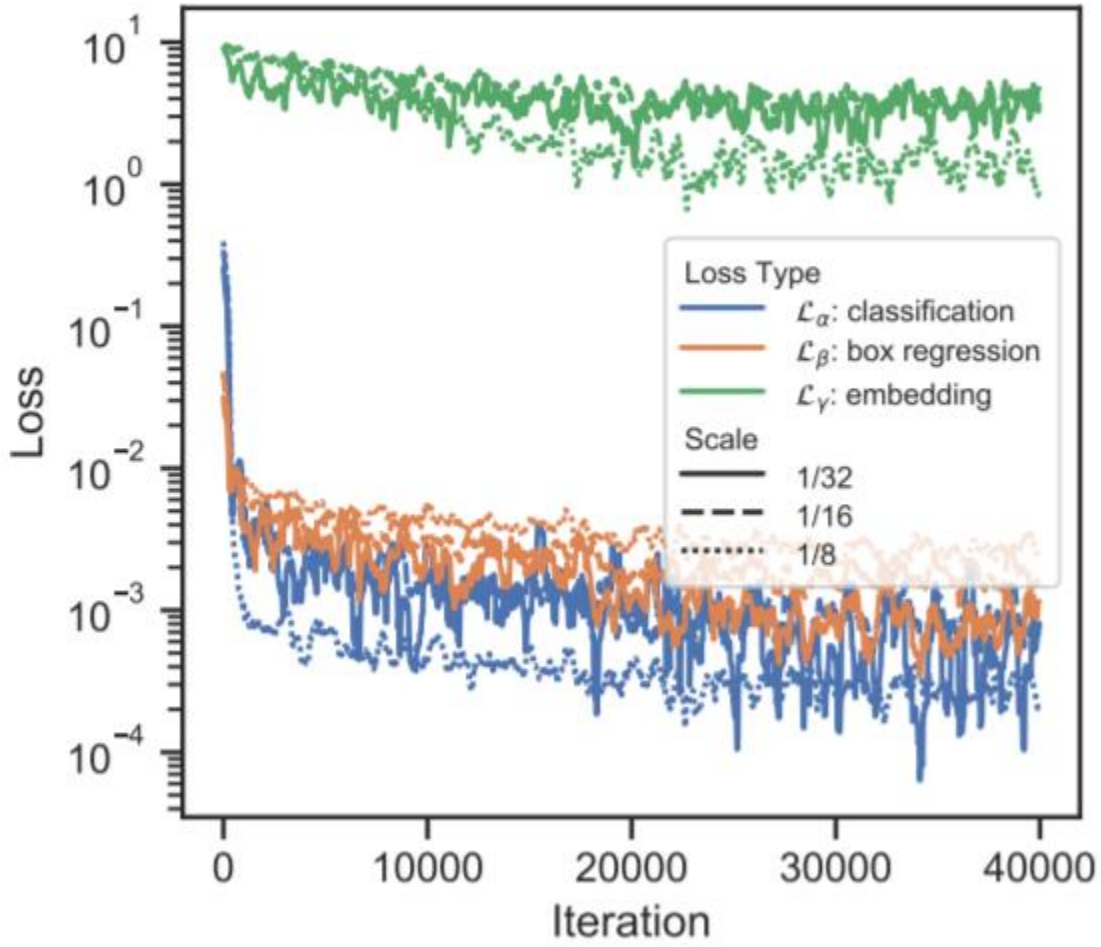
ID Sw.	lower	0	The total number of identity switches. Please note that we follow the stricter definition of identity switches as described in [3].

실험 결과 (로스 함수 비교)

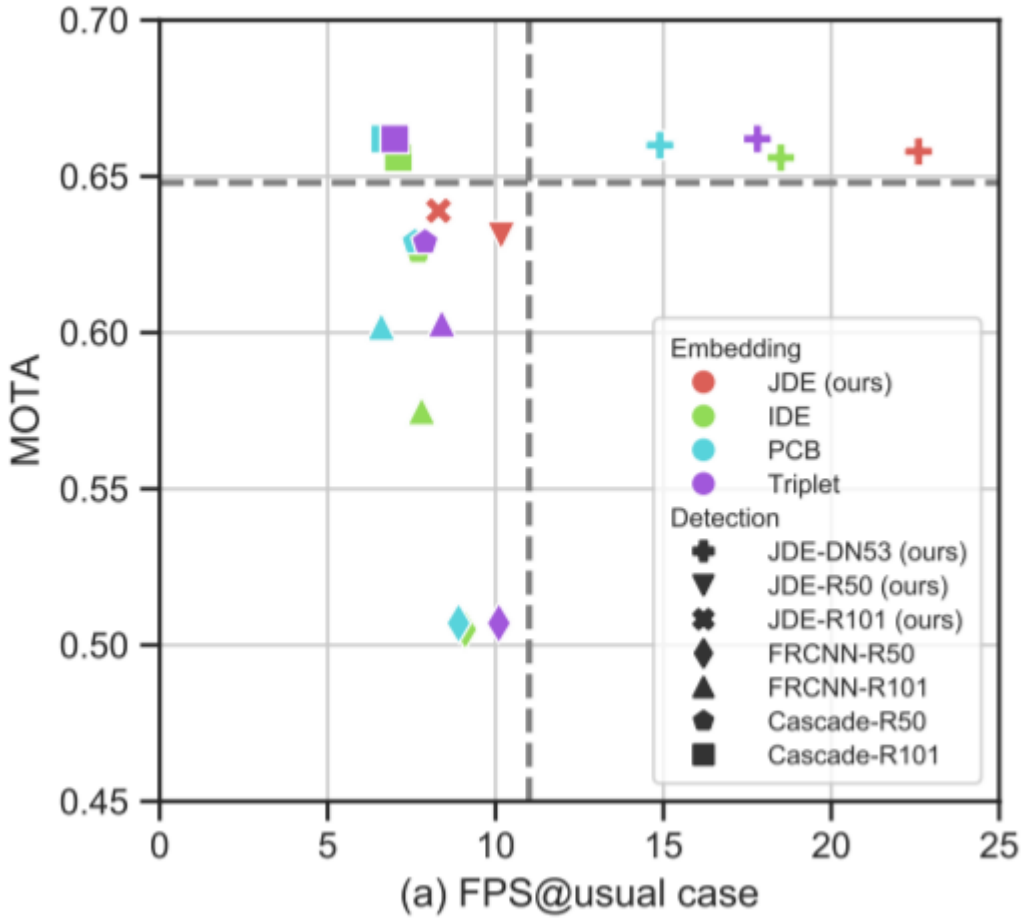
Embed. Loss	Weighting Strategy	Det	Emb	MOT	
		AP↑	TPR↑	MOTA↑	IDs↓
$\mathcal{L}_{triplet}$	App.Opt	81.6	42.2	59.5	375
$\mathcal{L}_{upper}$	App.Opt	81.7	44.3	59.8	346
$\mathcal{L}_{CE}$	App.Opt	<u>82.0</u>	88.2	<u>64.3</u>	<u>223</u>
$\mathcal{L}_{CE}$	Uniform	6.8	**94.8**	36.9	366
$\mathcal{L}_{CE}$	MGDA-UB	8.3	<u>93.5</u>	38.3	357
$\mathcal{L}_{CE}$	Loss.Norm	80.6	82.1	57.9	321
$\mathcal{L}_{CE}$	Uncertainty	**83.0**	90.4	**65.8**	**207**

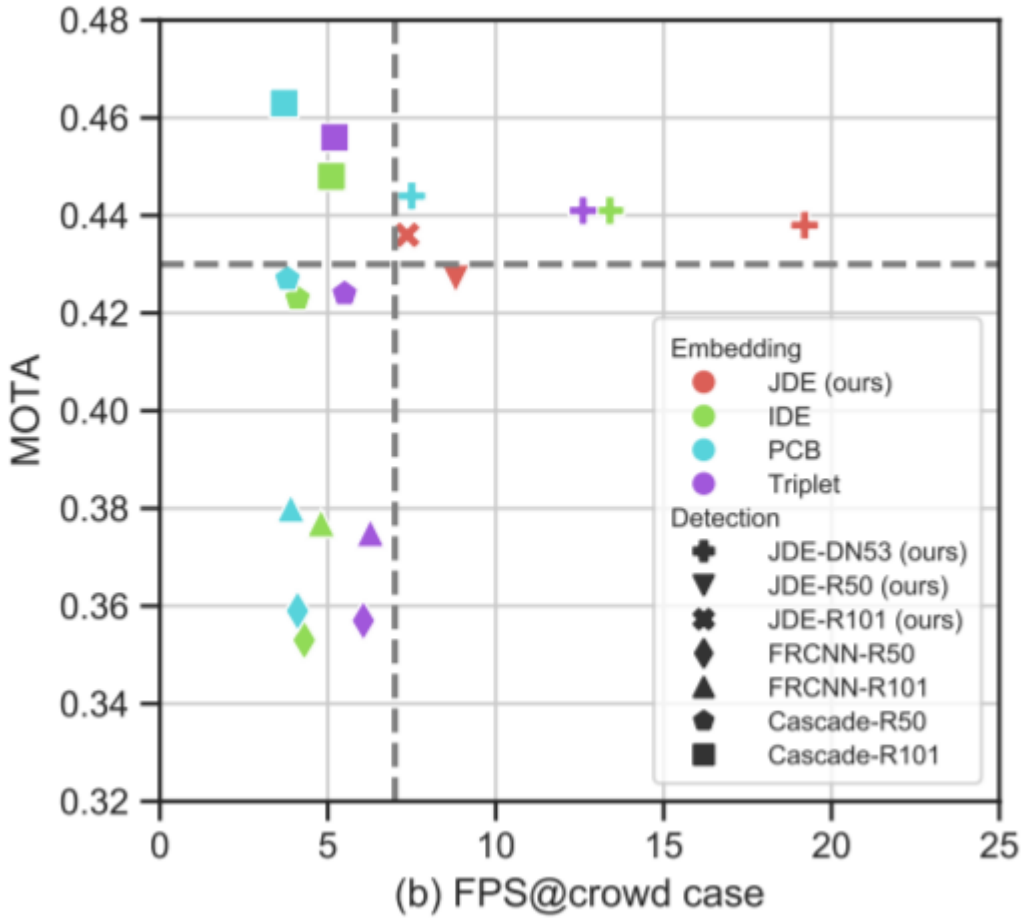
실험 결과 (가중치 전략 비교)





실험 결과 (추적 정확도 및 속도 비교)





실험 결과 (다른 방법들과 비교)

Method	Det	Emb	#box	#id	MOTA	IDF1	MT	ML	IDs	FPSD	FPSA	FPS
DeepSORT_2	FRCNN	WRN	429K	1.2k	61.4	62.2	32.8	18.2	781	<15*	17.4	<8.1
RAR16wVGG	FRCNN	Inception	429K	-	63.0	63.8	39.9	22.1	**482**	<15*	1.6	<1.5
TAP	FRCNN	MRCNN	429K	-	64.8	**73.5**	**40.6**	22.0	794	<15*	18.2	<8.2
CNNMTT	FRCNN	5-Layer	429K	0.2K	65.2	62.2	32.4	21.3	946	<15*	11.2	<6.4
POI	FRCNN	QAN	429K	16K	**66.1**	65.1	34.0	21.3	805	<15*	9.9	<6
JDE-864(ours)	JDE	-	270K	8.7K	62.1	56.9	34.4	**16.7**	1,608	**34.3**	81.0	**24.1**
JDE-1088(ours)	JDE	-	270K	8.7K	64.4	55.8	35.4	20.0	1,544	24.5	**81.5**	18.8

Q & A