DL Seminar

MOTDT

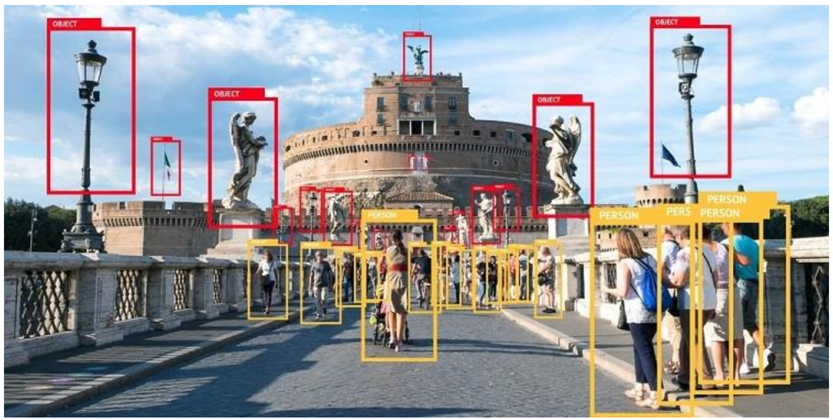
REAL-TIME MULTIPLE PEOPLE TRACKING WITH DEEPLY LEARNED CANDIDATE SELECTION AND PERSON RE-IDENTIFICATION

한양대학교
HANYANG UNIVERSITY

인공지능 연구실
김지성

Introduction

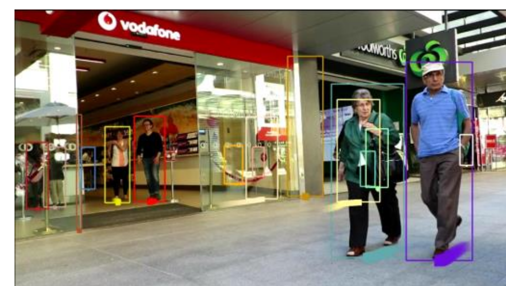
Detector, Tracker



Detector

Frame 1

Frame 2

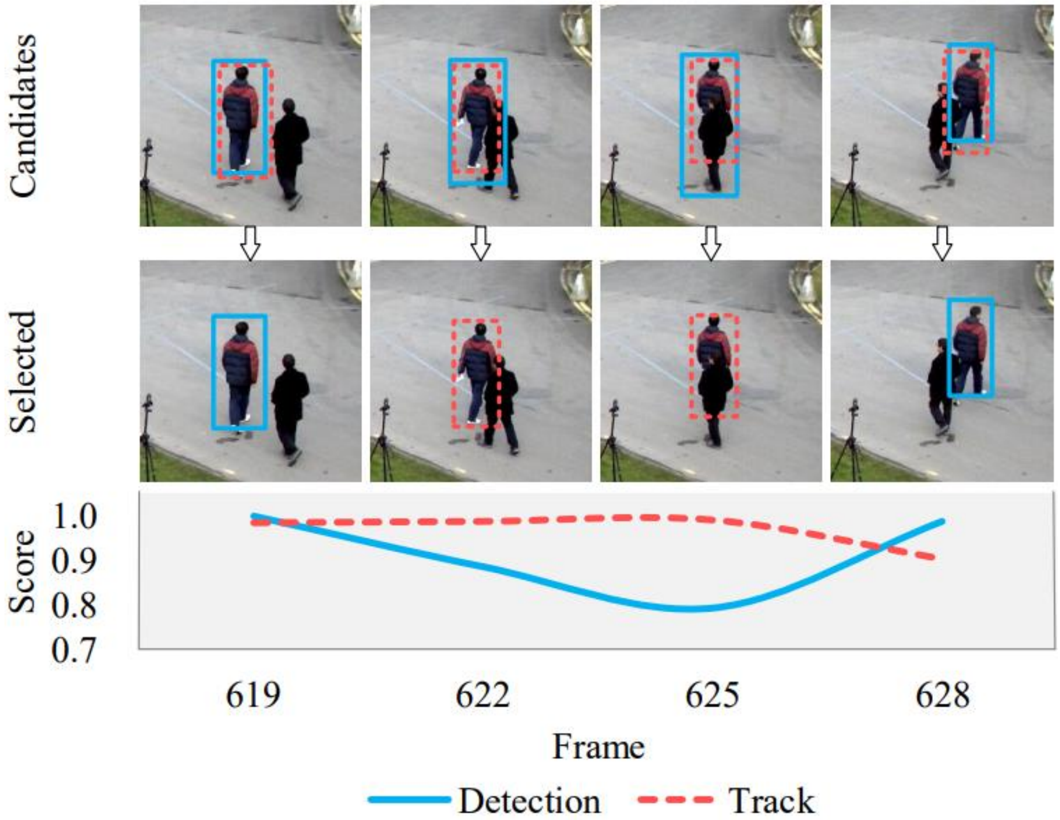


Frame 3

Tracker

Introduction

Candidates



Method

후보 필터링

Algorithm 1: The proposed tracking algorithm.

Input: A video sequence v with N_v frames and object detection $\{\mathcal{D}_k\}_{k=1}^{N_v}$

Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$; appearance of tracks $\mathcal{F}_{trk} \leftarrow \emptyset$

foreach *frame* f_k *in* v **do**

  Estimate score maps $\mathbf{z}$ from f using R-FCN

  `/* collect candidates */`

  $C_{det} \leftarrow \mathcal{D}_k$; $C_{trk} \leftarrow \emptyset$

  foreach t *in* $\mathcal{T}$ **do**

    Predict new location $\mathbf{x}^*$ of t using Kalman filter

    $C_{trk} \leftarrow C_{trk} \cup \{\mathbf{x}^*\}$

  end

  `/* select candidates */`

  $C \leftarrow C_{det} \cup C_{trk}$

  $S \leftarrow$ unified scores computed from Equation 3

  $C, S \leftarrow \mathrm{NMS}(C, S, \tau_{nms})$

  $C, S \leftarrow \mathtt{Filter}(C, S, \tau_s)$ `// filter out if` $s < \tau_s$

  `/* extract appearance features */`

  $\mathcal{F}_{det} \leftarrow \emptyset$

  foreach $\mathbf{x}$ *in* C_{det} **do**

    $\mathbf{I}_\mathbf{x} \leftarrow \mathtt{Crop}(f_k, \mathbf{x})$

    $\mathcal{F}_{det} \leftarrow \mathcal{F}_{det} \cup H_{reid}(\mathbf{I}_\mathbf{x})$

  end

  `/* hierarchical data association */`

  Associate $\mathcal{T}$ and C_{det} using distances of $\mathcal{F}_{trk}$ and $\mathcal{F}_{det}$

  Associate remaining tracks and candidates using IoU

  $\mathcal{F}_{trk} \leftarrow \mathcal{F}_{trk} \cup \mathcal{F}_{det}$

  `/* initialize new tracks */`

  $C_{remain} \leftarrow$ remaining candidates from C_{det}

  $\mathcal{F}_{remain} \leftarrow$ features of C_{remain}

  $\mathcal{T}, \mathcal{F}_{trk} \leftarrow \mathcal{T} \cup C_{remain}, \mathcal{F}_{trk} \cup \mathcal{F}_{remain}$

end

Object Detection

Kalman 예측 Track

(Detection + Tracking)후보집합 구성

Score 계산 -> 후보 필터링

Body Embedding

Body, IoU Matching

Tracklet Update

Tracking 결과 출력

Method

후보 필터링

후보 x에 대한 RoI 분류확률

$$p(y|\mathbf{z},\mathbf{x}) = \sigma\left(\frac{1}{wh}\sum_{i=1}^{k^2}\sum_{(x,y)\in bin_i}\mathbf{z}_i(x,y)\right), \qquad (1)$$

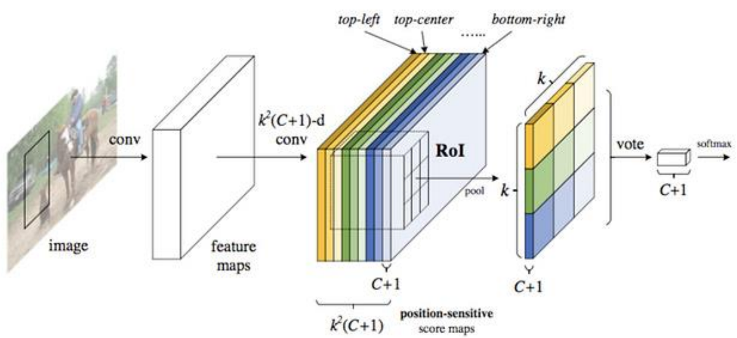


Figure 1: Key idea of **R-FCN** for object detection. In this illustration, there are $k \times k = 3 \times 3$ position-sensitive score maps generated by a fully convolutional network. For each of the $k \times k$ bins in an RoI, pooling is only performed on one of the k^2 maps (marked by different colors).

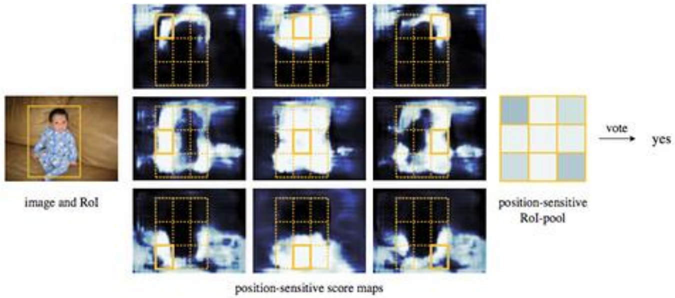


Figure 3: Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.

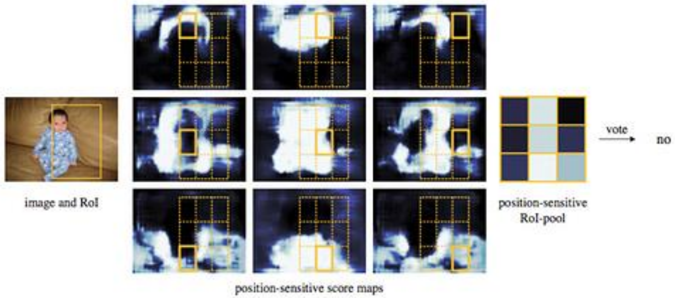


Figure 4: Visualization when an RoI does not correctly overlap the object.

Method

후보 필터링

Tracklet 신뢰도

$$s_{trk} = \max(1 - log(1 + \alpha \cdot L_{trk}), 0) \cdot \mathbb{1}(L_{det} \geq 2), \quad (2)$$

Ltrk : 최종적으로 트래킹한 결과의 수
Ldet : 최종적으로 탐지한 결과의 수

적어도 탐지된 결과가 2개 이상이고, 최종 트래킹 수가 적을수록 신뢰도가 높다

후보가 많을 수록 신뢰도가 낮음

후보가 적을 수록 신뢰도가 높음

Method

후보 필터링

후보 x에 대한 score 계산

$$s = p(y|\mathbf{z}, \mathbf{x}) \cdot (\mathbb{1}(\mathbf{x} \in C_{det})) + s_{trk}\mathbb{1}(\mathbf{x} \in C_{trk})). \qquad (3)$$

X : 후보
P(y|z, x) : 식 1에서 계산된 신뢰도
Strk : 식 2에서 계산된 신뢰도

R-FCN으로 계산한 점수 + Tracklet(후보) 신뢰도

만약 최종 신뢰도 s가 0.4 이하라면 후보를 버린다.

Method

Body Embedding

Algorithm 1: The proposed tracking algorithm.

Input: A video sequence v with N_v frames and object detection $\{\mathcal{D}_k\}_{k=1}^{N_v}$
Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$; appearance of tracks $\mathcal{F}_{trk} \leftarrow \emptyset$
foreach *frame* f_k *in* v **do**
  Estimate score maps $\mathbf{z}$ from f using R-FCN
  `/* collect candidates */`
  $C_{det} \leftarrow \mathcal{D}_k$; $C_{trk} \leftarrow \emptyset$
  foreach t *in* $\mathcal{T}$ **do**
    Predict new location $\mathbf{x}^*$ of t using Kalman filter
    $C_{trk} \leftarrow C_{trk} \cup \{\mathbf{x}^*\}$
  end
  `/* select candidates */`
  $C \leftarrow C_{det} \cup C_{trk}$
  $S \leftarrow$ unified scores computed from Equation 3
  $C, S \leftarrow \mathtt{NMS}(C, S, \tau_{nms})$
  $C, S \leftarrow \mathtt{Filter}(C, S, \tau_s)$ `// filter out if` $s < \tau_s$
  `/* extract appearance features */`
  $\mathcal{F}_{det} \leftarrow \emptyset$
  foreach $\mathbf{x}$ *in* C_{det} **do**
    $\mathbf{I}_\mathbf{x} \leftarrow \mathtt{Crop}(f_k, \mathbf{x})$
    $\mathcal{F}_{det} \leftarrow \mathcal{F}_{det} \cup H_{reid}(\mathbf{I}_\mathbf{x})$
  end
  `/* hierarchical data association */`
  Associate $\mathcal{T}$ and C_{det} using distances of $\mathcal{F}_{trk}$ and $\mathcal{F}_{det}$
  Associate remaining tracks and candidates using IoU
  $\mathcal{F}_{trk} \leftarrow \mathcal{F}_{trk} \cup \mathcal{F}_{det}$
  `/* initialize new tracks */`
  $C_{remain} \leftarrow$ remaining candidates from C_{det}
  $\mathcal{F}_{remain} \leftarrow$ features of C_{remain}
  $\mathcal{T}, \mathcal{F}_{trk} \leftarrow \mathcal{T} \cup C_{remain}, \mathcal{F}_{trk} \cup \mathcal{F}_{remain}$
end

Method

Body Embedding

128d 벡터

Euclidean distance

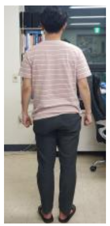
A인물의 사진1

FaceNet

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,-0.11010353, 0.02105946,-0.09806988,
-0.09223023,-0.01714757,-0.16107245,-0.04865564, 0.05062422,
0.04144309,-0.03346116,-0.03011878, 0.15263124, 0.01069992,
-0.23673587, 0.05740198, 0.03050802, 0.23846291, 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, 0.03233573,
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,-0.04398936, 0.08519959, 0.14640823, 0.00993889,
0.02739849,-0.11102622, 0.01848426, 0.23328975,-0.11351117,
-0.04641433, 0.22538319,-0.00492372, 0.10828383, 0.02823116,
0.02502055,-0.02946196, 0.0702677 , -0.09549975,-0.0334047 ,
0.00996118,-0.08729131,-0.04567208, 0.09973253,-0.14260028,
0.10422572,-0.00379006, 0.05201333, 0.02785442,-0.05933321,
-0.09983464,-0.02418252, 0.13822252,-0.24259827, 0.2536535 ,
0.12104575, 0.14091188, 0.07011457, 0.10865125, 0.04752614,
0.02150964,-0.04581762,-0.23496597,-0.01055925, 0.11252803,
-0.05433004, 0.10194337,-0.02596316])

0.78

1.33



A인물의 사진2

FaceNet

array([-0.08902847, 0.14762324, 0.05718813,-0.05012994,-0.09676401,
0.02028476,-0.08940892,-0.0386399 , 0.10385386,-0.031321 ,
0.2729257 , -0.02359607,-0.23816103,-0.09732635,-0.03333118,
0.1333721 , -0.19538365,-0.07668579, 0.00275577, 0.03931012,
0.15097558, 0.08741103, 0.0678171 , 0.01446525,-0.09825172,
-0.33090472,-0.06750867,-0.10570621, 0.01662679,-0.11775655,
-0.10916033,-0.04665562,-0.1787585 , -0.05166516, 0.04875493,
0.0426253 , -0.0591871 , -0.04663718, 0.16380328, 0.00791238,
-0.22012679, 0.04296945, 0.04917528, 0.23990902, 0.21356051,
0.00606929, 0.00836444,-0.13134141, 0.09845343,-0.162596 ,
0.11051578, 0.15520622, 0.12857951, 0.05687075, 0.01556353,
-0.09330104,-0.04209765, 0.14811578,-0.10172325, 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,-0.17039801,-0.25199199,
0.0466072 , 0.38125145, 0.03306871,-0.19173753, 0.01305585,
-0.07093169,-0.05639979, 0.07696941, 0.15101001, 0.02203412,
0.0188837 , -0.10816664, 0.04892536, 0.23276457,-0.10933174,
-0.06033619, 0.2151441 , -0.01290991, 0.10794014, 0.02028765,
0.05325544,-0.03414371, 0.055746 , -0.10889978,-0.02816046,
0.01456824,-0.07452301,-0.06755616, 0.11638117,-0.13528842,
0.10086828, 0.00977207, 0.03836292, 0.02850841,-0.05801099,
-0.10640397,-0.03940248, 0.12043379,-0.25706992, 0.24780291,
0.14584672, 0.13685006, 0.05027201, 0.09039105, 0.03249723,
0.04752009,-0.02310574,-0.21513188,-0.00798578, 0.09137717,
-0.01307906, 0.06952387,-0.03653527])

1.27

B인물의 사진1

FaceNet

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,-0.11010353, 0.02105946,-0.09806988,
-0.09223023,-0.01714757,-0.16107245,-0.04865564, 0.05062422,
0.04144309,-0.03346116,-0.03011878, 0.15263124, 0.01069992,
-0.23673587, 0.05740198, 0.03050802, 0.23846291, 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, 0.03233573,
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,-0.04398936, 0.08519959, 0.14640823, 0.00993889,
0.02739849,-0.11102622, 0.01848426, 0.23328975,-0.11351117,
-0.04641433, 0.22538319,-0.00492372, 0.10828383, 0.02823116,
0.02502055,-0.02946196, 0.0702677 , -0.09549975,-0.0334047 ,
0.00996118,-0.08729131,-0.04567208, 0.09973253,-0.14260028,
0.10422572,-0.00379006, 0.05201333, 0.02785442,-0.05933321,
-0.09983464,-0.02418252, 0.13822252,-0.24259827, 0.2536535 ,
0.12104575, 0.14091188, 0.07011457, 0.10865125, 0.04752614,
0.02150964,-0.04581762,-0.23496597,-0.01055925, 0.11252803,
-0.05433004, 0.10194337,-0.02596316])

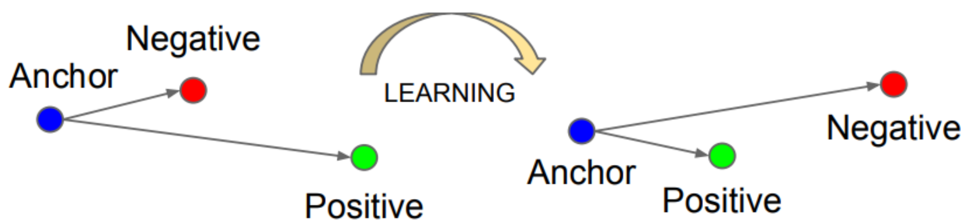
Method

Body Embedding - Triplet

$$l_{triplet} = \frac{1}{N} \sum_{\langle \mathbf{I}_i, \mathbf{I}_j, \mathbf{I}_k \rangle \in \mathbf{T}} \max(d_{ij} - d_{ik} + m, 0), \quad (4)$$

Triplet : 세 개의 데이터

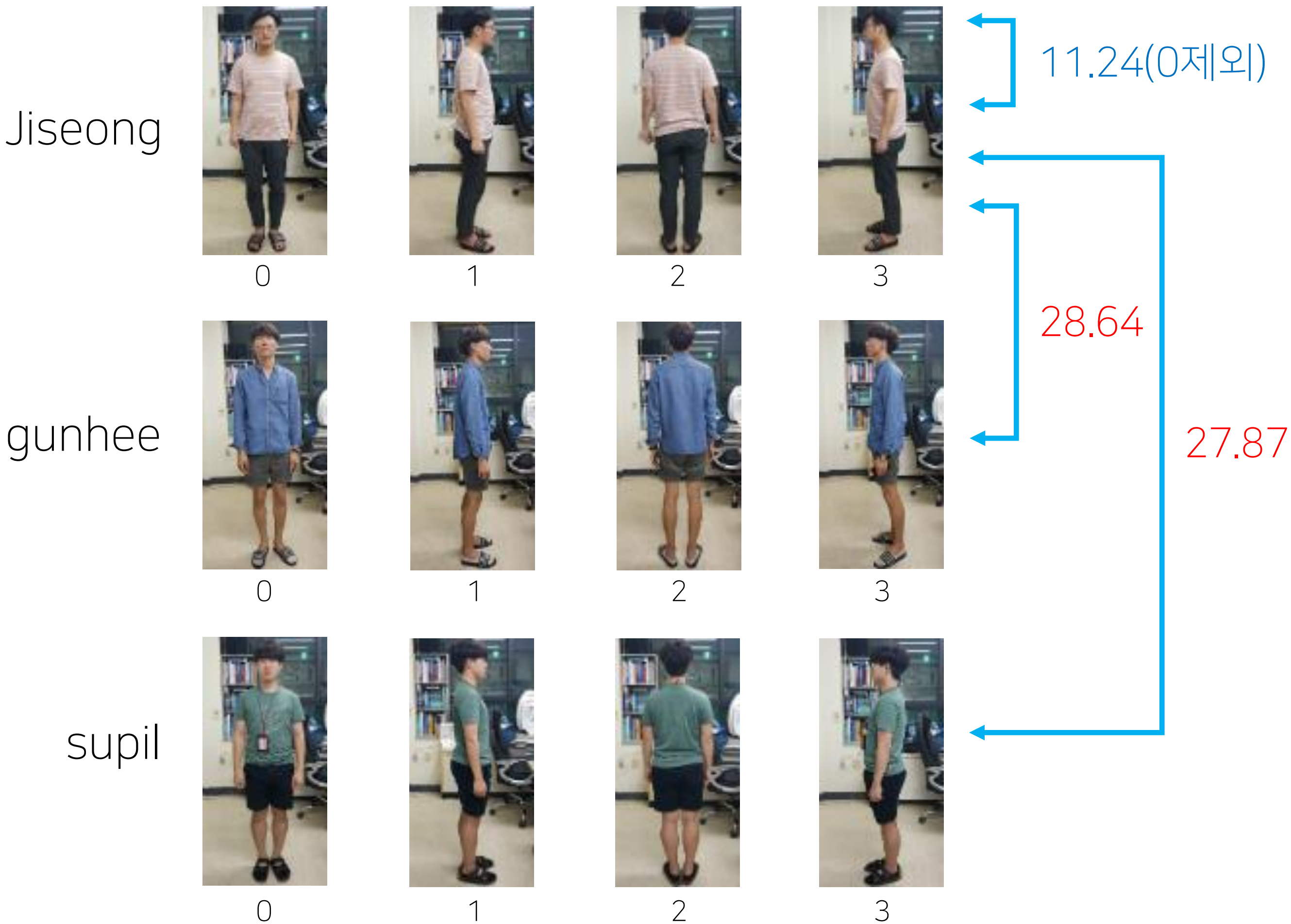
- Anchor(x_i^a) : 기준 인물의 벡터
- Positive(x_i^p) : 기준과 같은 인물의 벡터
- Negative(x_i^n) : 기준과 다른 인물의 벡터



기준 인물과 같은 인물은 가깝도록, 기준 인물과 다른 인물은 멀도록

Person Re-Identification

평균 거리




```
jiseong 0, jiseong 0 : 0
jiseong 0, jiseong 1 : 14.6833314343748
jiseong 0, jiseong 2 : 7.14393362143874
jiseong 0, jiseong 3 : 13.7174603497803
jiseong 1, jiseong 0 : 14.6833314343748
jiseong 1, jiseong 1 : 0
jiseong 1, jiseong 2 : 13.3976718560914
jiseong 1, jiseong 3 : 5.4062859718527
jiseong 2, jiseong 0 : 7.14393362143874
jiseong 2, jiseong 1 : 13.3976718560914
jiseong 2, jiseong 2 : 0
jiseong 2, jiseong 3 : 13.1192839922723
jiseong 3, jiseong 0 : 13.7174603497803
jiseong 3, jiseong 1 : 5.4062859718527
jiseong 3, jiseong 2 : 13.1192839922723
jiseong 3, jiseong 3 : 0
평균거리 : 11.2446612043017

jiseong 0, gunhee0 : 33.702875406024
jiseong 0, gunhee1 : 30.6240802173837
jiseong 0, gunhee2 : 34.6539498741167
jiseong 0, gunhee3 : 30.0388592472747
jiseong 1, gunhee0 : 28.4433327086116
jiseong 1, gunhee1 : 24.1252707695199
jiseong 1, gunhee2 : 29.460226635402
jiseong 1, gunhee3 : 23.97228405069
jiseong 2, gunhee0 : 31.3361323432144
jiseong 2, gunhee1 : 28.299821729878
jiseong 2, gunhee2 : 32.1328332890316
jiseong 2, gunhee3 : 27.8813871223626
jiseong 3, gunhee0 : 27.7410707429242
jiseong 3, gunhee1 : 23.9328520321636
jiseong 3, gunhee2 : 28.771589169261
jiseong 3, gunhee3 : 23.1914684226621
평균거리 : 28.6442521100325

jiseong 0, supil0 : 32.5034984023311
jiseong 0, supil1 : 31.0123639922488
jiseong 0, supil2 : 33.1022183632668
jiseong 0, supil3 : 28.2895292724031
jiseong 1, supil0 : 26.9324479963173
jiseong 1, supil1 : 25.3354276975164
jiseong 1, supil2 : 28.4278065042272
jiseong 1, supil3 : 22.5782505170552
jiseong 2, supil0 : 31.1035040681003
jiseong 2, supil1 : 30.1407394004526
jiseong 2, supil2 : 31.6739051391745
jiseong 2, supil3 : 27.3928874045504
jiseong 3, supil0 : 25.716432805367
jiseong 3, supil1 : 23.8073005976364
jiseong 3, supil2 : 27.0416266409934
jiseong 3, supil3 : 20.8900794862151
평균거리 : 27.871751142991
```

Method

Matching

Algorithm 1: The proposed tracking algorithm.

Input: A video sequence v with N_v frames and object detection $\{\mathcal{D}_k\}_{k=1}^{N_v}$

Output: Tracks $\mathcal{T}$ of the video

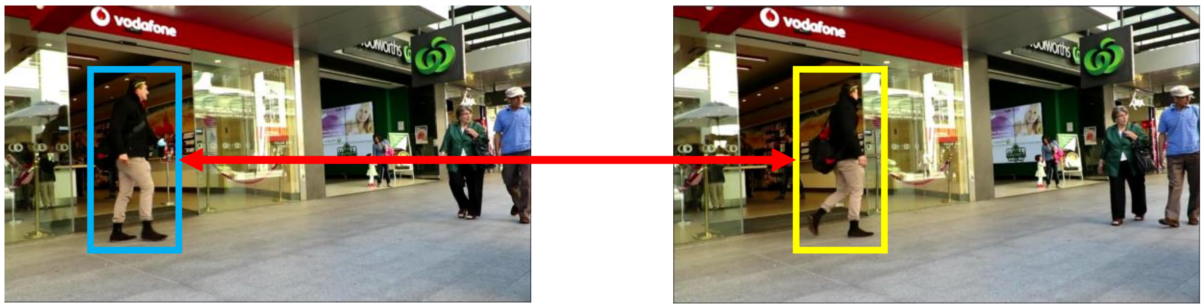
Initialization: $\mathcal{T} \leftarrow \emptyset$; appearance of tracks $\mathcal{F}_{trk} \leftarrow \emptyset$
foreach *frame* f_k *in* v **do**
  Estimate score maps $\mathbf{z}$ from f using R-FCN
  `/* collect candidates */`
  $C_{det} \leftarrow \mathcal{D}_k$; $C_{trk} \leftarrow \emptyset$
  foreach t *in* $\mathcal{T}$ **do**
    Predict new location $\mathbf{x}^*$ of t using Kalman filter
    $C_{trk} \leftarrow C_{trk} \cup \{\mathbf{x}^*\}$
  end
  `/* select candidates */`
  $C \leftarrow C_{det} \cup C_{trk}$
  $S \leftarrow$ unified scores computed from Equation 3
  $C, S \leftarrow \texttt{NMS}(C, S, \tau_{nms})$
  $C, S \leftarrow \texttt{Filter}(C, S, \tau_s)$ `// filter out if` $s < \tau_s$
  `/* extract appearance features */`
  $\mathcal{F}_{det} \leftarrow \emptyset$
  foreach $\mathbf{x}$ *in* C_{det} **do**
    $\mathbf{I}_\mathbf{x} \leftarrow \texttt{Crop}(f_k, \mathbf{x})$
    $\mathcal{F}_{det} \leftarrow \mathcal{F}_{det} \cup H_{reid}(\mathbf{I}_\mathbf{x})$
  end
  `/* hierarchical data association */`
  Associate $\mathcal{T}$ and C_{det} using distances of $\mathcal{F}_{trk}$ and $\mathcal{F}_{det}$
  Associate remaining tracks and candidates using IoU
  $\mathcal{F}_{trk} \leftarrow \mathcal{F}_{trk} \cup \mathcal{F}_{det}$
  `/* initialize new tracks */`
  $C_{remain} \leftarrow$ remaining candidates from C_{det}
  $\mathcal{F}_{remain} \leftarrow$ features of C_{remain}
  $\mathcal{T}, \mathcal{F}_{trk} \leftarrow \mathcal{T} \cup C_{remain}, \mathcal{F}_{trk} \cup \mathcal{F}_{remain}$
end

Method

Matching

1. 후보와 Active tracklet을 Body Feture를 이용하여 매칭

이전 프레임 현재 프레임



Active Tracklet

후보

Body Feature Threshold : 0.64
IoU Threshold : 0.4

Method

Matching

2. 후보와 Lost Tracklet과 Body Feature를 이용하여 매칭

이전 프레임 1 | 이전 프레임 2 | 현재 프레임

Tracking Object를 잃어버림

- Active Tracklet
- Lost Tracklet
- 후보

Body Feature Threshold : 0.64
IoU Threshold : 0.4

Method

Matching

3. 후보와 Active tracklet을 IoU를 이용하여 매칭

이전 프레임

현재 프레임

Active Tracklet

후보

IoU Threshold : 0.7

Method

Matching

4. 후보와 Lost Tracklet과 IoU를 이용하여 매칭

이전 프레임 1

이전 프레임 2

Tracking Object를 잃어버림

현재 프레임

- Active Tracklet
- Lost Tracklet
- 후보

IoU Threshold : 0.7

Method

MOTA

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

Ground Truth와 관심영역의 IOU가 0.5 이상일 때 True로 판단

MOTA : tracker 성능지표로 적합

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

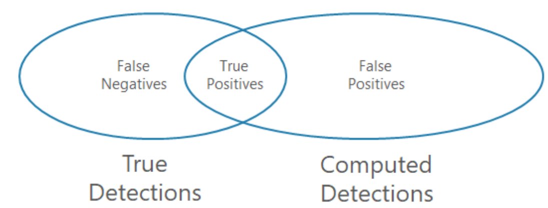
t : 프레임 인덱스
FN : 잘못탐지
FP : 놓친 객체
IDSW : id가 바뀐 횟수
GT : Ground Truth의 수

Method

IDF1

IDF1 : 얼마나 지속적으로 객체를 동일하게 판단하는가

- ID Precision $P = \frac{TP}{TP+FP} = \frac{TP}{C}$
- ID Recall $R = \frac{TP}{TP+FN} = \frac{TP}{T}$
- F_1-score $F_1 = 2\frac{PR}{P+R} = \frac{TP}{\frac{T+C}{2}}$



TP : 가장 많이 나온 ID의 개수 = 12

FP : 나머지 ID의 개수 = 2

FN : True – TP = 4

Vision Module

MOT Challenge

Method	C	T	A	MOTA↑	IDF1↑	IDS↓	FAF↓
Baseline				28.4	32.8	628	0.85
	✓			33.0	37.6	445	0.77
	✓	✓		33.7	37.3	475	0.63
			✓	30.6	42.4	234	1.01
Proposed	✓	✓	✓	**35.7**	**45.3**	**184**	**0.58**

Method	Length	MOTA↑	IDF1↑	IDS↓	FAF↓
None	-	33.7	37.3	475	0.63
Color histogram	750	34.9	38.6	250	0.73
HOG	1152	34.6	38.5	317	0.70
Color + HOG	1902	34.7	39.3	307	0.68
ReID feature	**512**	**35.7**	**45.3**	**184**	**0.58**

Tracker	Method	MOTA(%)↑	IDF1(%)↑	IDR(%)↑	MT(%)↑	ML(%)↓	FP↓	FN↓	IDS↓	FPS↑
LINF1 [2]	batch	41.0	45.7	34.2	11.6	51.3	7,896	99,224	**430**	**4.2**
MHT_DAM [5]	batch	45.8	46.1	35.3	16.2	43.2	6,412	91,758	590	0.8
JMC [11]	batch	46.3	46.3	35.6	15.5	**39.7**	**6,373**	90,914	657	0.8
LMP [6]	batch	**48.8**	**51.3**	**40.1**	**18.2**	40.1	6,654	**86,245**	481	0.5
EAMTT [7]	**online**	38.8	42.4	31.5	7.9	49.1	8,114	102,452	965	11.8
CDA_DDAL [1]	**online**	43.9	45.1	34.1	10.7	44.4	6,450	95,175	676	0.5
STAM [8]	**online**	46.0	50.0	38.5	14.6	43.6	6,895	91,117	**473**	0.2
AMIR [3]	**online**	47.2	46.3	34.8	14.0	41.6	**2,681**	92,856	774	1.0
MOTDT (Ours)	**online**	**47.6**	**50.9**	**40.3**	**15.2**	**38.3**	9,253	**85,431**	792	**20.6**

감사합니다.