IRIM at TRECVID 2017: Instance Search

Presenter : Pierre-Etienne Martin

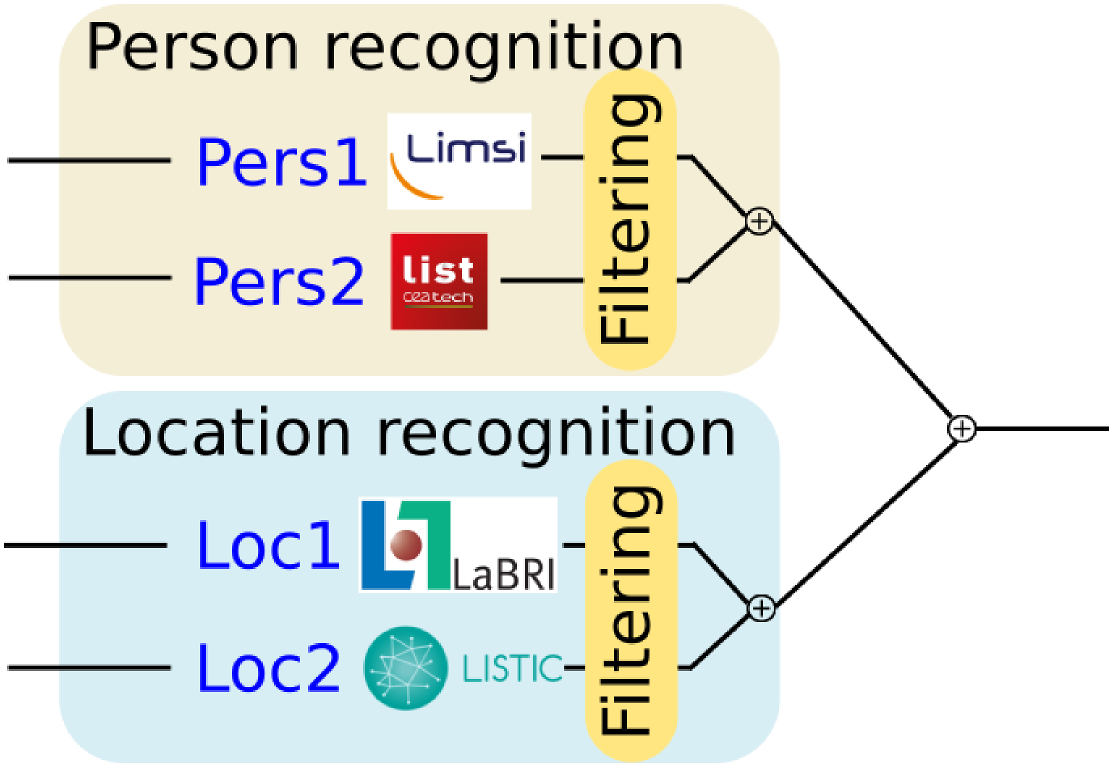
Boris Mansencal, Jenny Benois-Pineau – **LaBRI**
Hervé Bredin - **LIMSI**
Alexandre Benoit, Nicolas Voiron, Patric Lambert – **LISTIC**
Hervé Le Borgne, Adrian Popescu, Alexandru L. Ginsca – **CEA LIST**
Georges Quénot - **LIG**

IRIM

- **Consortium of French teams** working on Multimedia Indexing and Retrieval, coordinated by Georges Quénot, **LIG**.
- Long-time participant (2007-2012: HLFE, 2013-2015: SIN, 2011-2014, 2016-2017: INS)
- Also individual members participations (SBD, Rushes, Copy Detection, …)
- **INS2017:** participation of **four French laboratories: CEA LIST, LaBRI, LIMSI, LISTIC**, coordinated by LaBRI.

Proposed approach

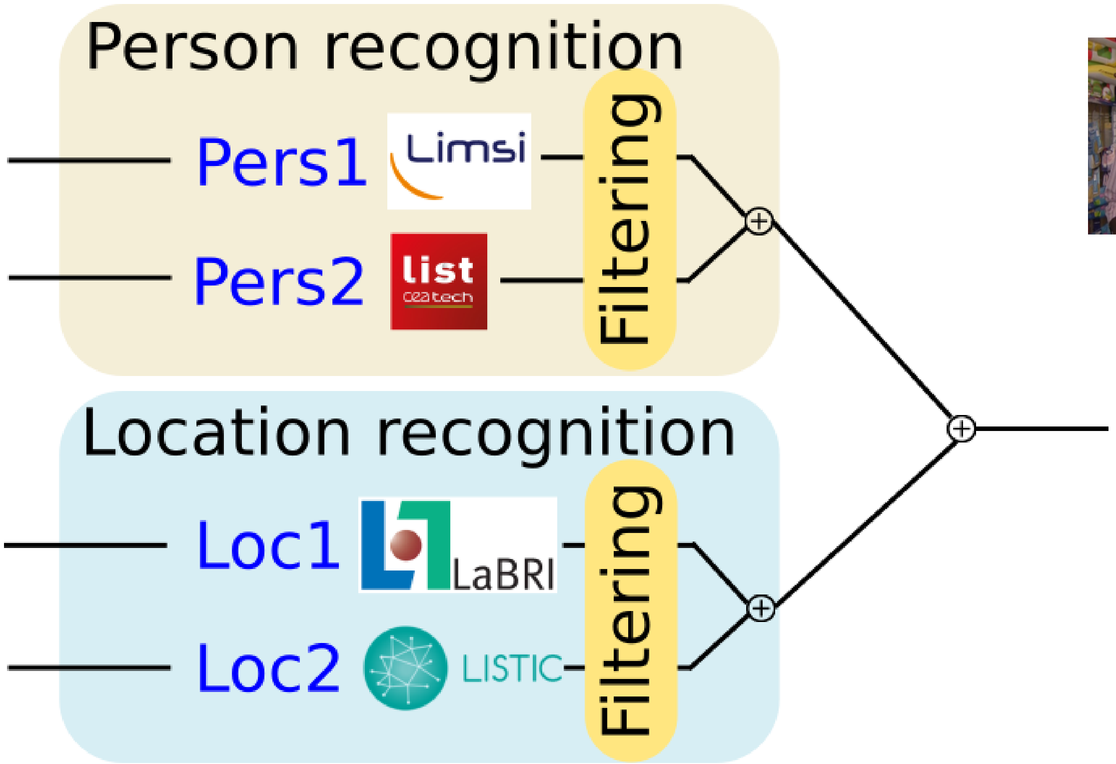
Late fusion of individual methods



Dot at the market

Proposed approach

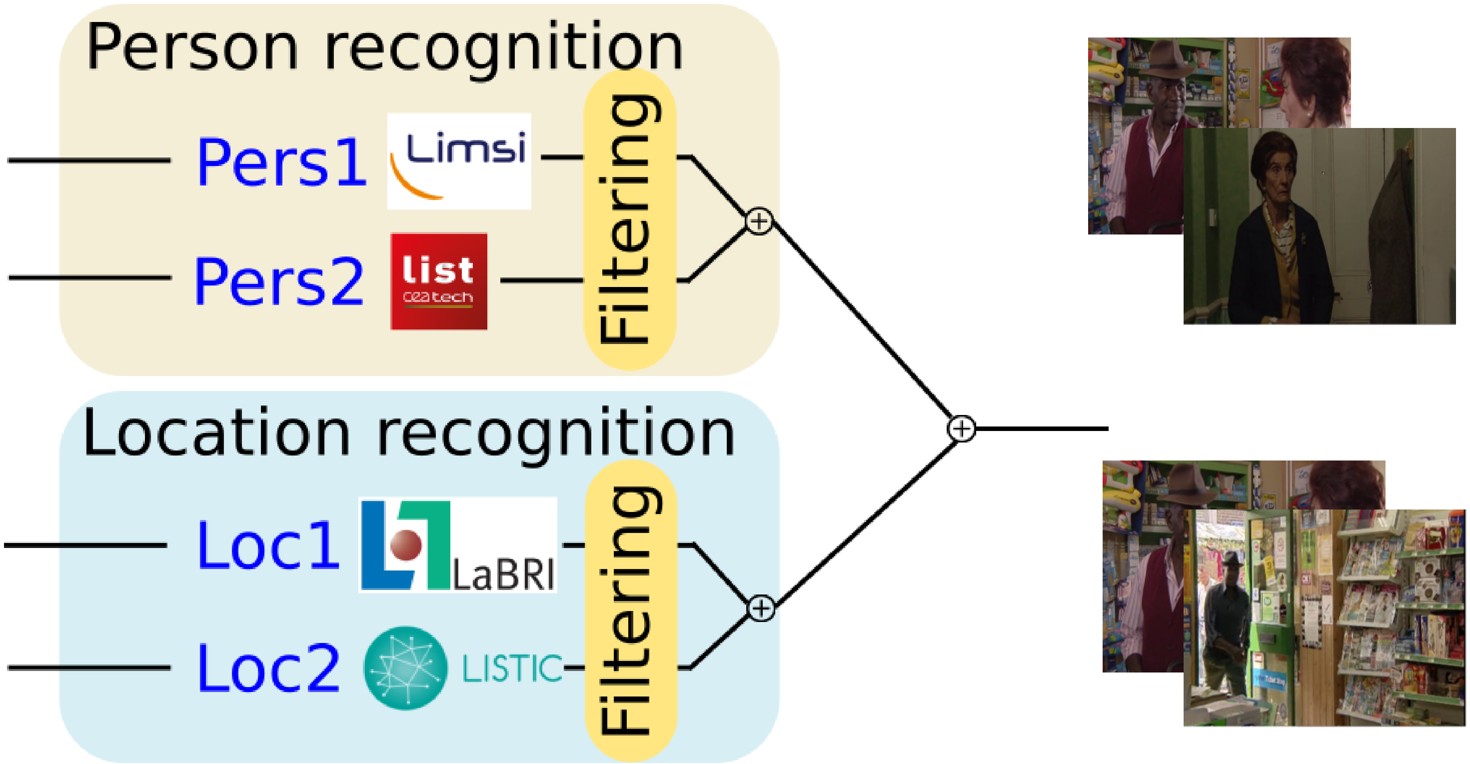
Late fusion of individual methods



Dot at the market

Proposed approach

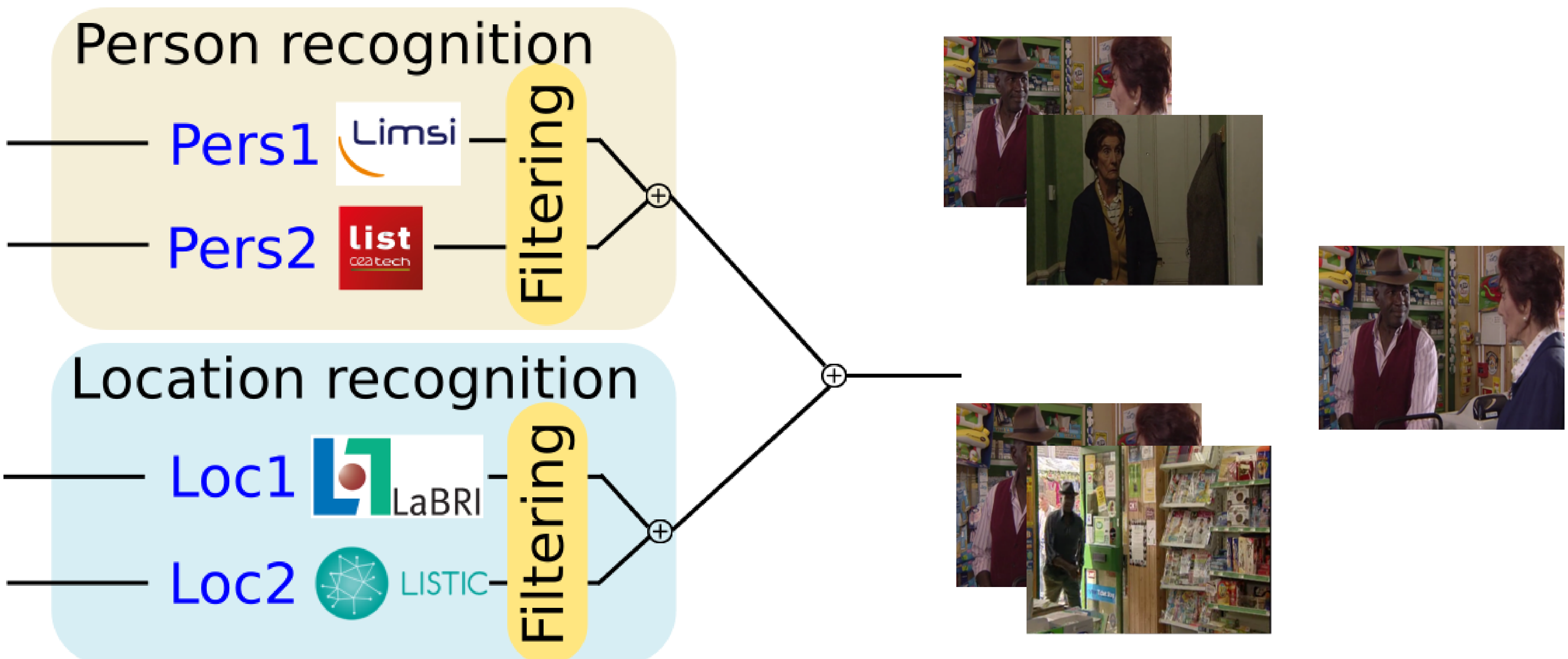
Late fusion of individual methods



Dot at the market

Proposed approach

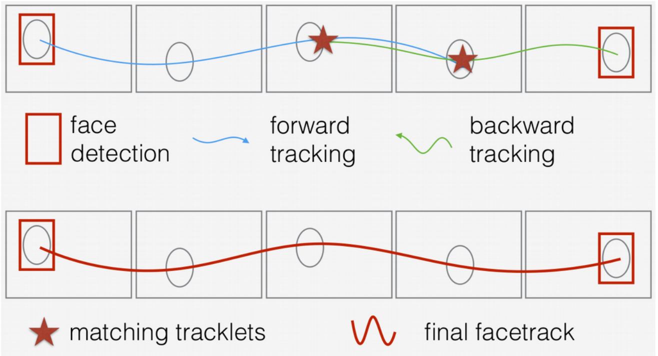
Late fusion of individual methods



Dot at the market

Person recognition Pers1

- Shot boundaries:
  Optical flow + displaced frame difference
- Face tracking-by-detection[1,2]:
  HOG detector (@ 2fps) + correlation tracker
- Face description:
  ResNet pre-trained on FaceScrub & VGG-Face (99.38% on LFW)
  Descriptors: 128 D
  Average for each face track
  Comparaison: Euclidean distance



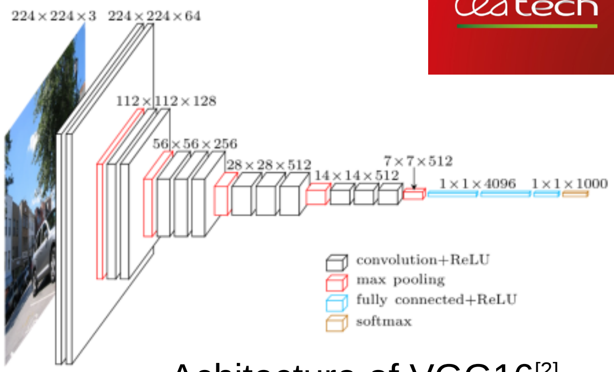
Face tracking-by-detection

[1] H. Bredin « Pyannote-video: Face Detection, Tracking and Clustering in Videos »
http://github.com/pyannote/pyannote-video
[2] dlib.net

Person recognition Pers2

- Face detection:
  Viola-Jones [OpenCV] (front and profile)

- Face description:
  FC7 of a VGG16 network[1]
  Model trained on external database
  → 5000 ids, ~800 images/id, 98.6% on LFW[3]



Achitecture of VGG16[2]

- Query expansion[4]:
  Images collected automatically from YouTube/Google/Bing
  kNN-based re-ranking

- Coherency criterion:
  K nearest neigborhood (K=4)

[1] Y. Tamaazousti *et al*., « Vision-language integration using contrained local semantic features » CVIU 2017
[2] Leonard Blier, « A brief report of the Heuritech Deep Learning Meetup #5 », 29 Feb. 2016, heuritech.com
[3] Labeled Faces in the Wild, http://vis-www.cs.umass.edu/lfw/
[4] P.D. Vo *et al*., « Harnessing noisy web images for deep representation », CVIU 2017

Location recognition Loc1

- BoW: (@ 1fps)
  Keypoints: Harris-Laplace detector
  Desciptors: OpponentSIFT → RootSIFT
  Clustering: 1M words using approximate K-means algorithm
  Weighted: Tf-idf scheme[1]
  Normalization: L2-norm
  Comparaison: Cosine similarity

- Filter out:
  Keypoints on characters bounding boxes
  computed from face tracks

- Option: Fast re-ranking[2]
  Geometric verification using Ransac
  Use words instead of descriptors for matching

Example of filtering

[1] M. J. Salton, G; McGill, Introduction to modern information retrieval. McGraw-Hill, 1986.
[2] X. Zhou *et al.*, « A practical spacial re-ranking method for instance search from videos » ICIP2014

Location recognition Loc2

- Pretrained GoogLeNet Places365[1]

- Features:

  Output of the pool5/7x7_s1 layer (last layer before classification)

- Similarity score between features:

$$Sim(s,l) = \exp\left(\frac{minDistLocation(s,l)}{topicsDistStd}\right)$$

with l the locations (6-12 frames)
s the shot (10 frames extracted)
average over the 10 frames

[1] https://github.com/CSAILVision/places365

Filtering 1/3

- Credits shots filtering

  Filters out shots before opening credits (before frame 3500) and after end credits (97% of length movie) by near duplicate frame detection

<u>Last image of opening credits</u>

<u>First image of end credits</u>

Filtering 2/3

- Indoor/Outdoor shots filtering

  Pretrained VGG Places365[1]: 365 categories manually classified as indoor & outdoor (190 indoors, 175 outdoors)

  /a/airfield 0
  /a/airplane_cabin 1
  /a/airport_terminal 1
  /a/alcove 1
  /a/alley 0
  /a/amphitheater 0
  /a/amusement_arcade 1
  /a/apartment_building/outdoor 0
  …

  Sum the K = 5 best probabilities over Indoors (1) and Outdoors (0)

[1] https://github.com/CSAILVision/places365

Filtering 3/3

- Shots threads filtering
  - Temporally constrained clustering (K=5 clusters neighborhood)
  - Uses BoW signature :

$$Inter_k = Signature(Shot_n) \cap Signature(Shot_k)$$

$$if \max_{k \in NC}(Inter_k) > Threshold$$

$$then \; Shot_n \in C_{Shot_i} \; with \; i = argMax(Inter_k)$$

Late fusion

- Fusion using the rank:

Fusion 1:

$$\Theta(\text{rank1}, \text{rank2}) = \alpha * \text{rank1} + (1 - \alpha) * \text{rank2}$$

Fusion 2:

$$\Phi(\text{rank1}, \text{rank2}) = \alpha * \text{sig}(\text{rank1}) + (1 - \alpha) * \text{sig}(\text{rank2})$$

Runs

31 fully automatic runs submitted by 7 participants
6 first runs by PKU/ICST, IRIM 2nd / 7 participants

Notations:

C:	Credits filtering	p1 = pers1 + T	Θ: late fusion 1
I:	Indoor/outdoor filtering	p2 = pers2 + T	Φ: late fusion 2
T:	Shots threads filtering	l1 = loc1 + C + I + R + T	E: E conditions
R:	Fast re-ranking	l2 = loc2 + C + I + T	A : A conditions

Runs

31 fully automatic runs submitted by 7 participants
6 first runs by PKU/ICST, IRIM 2nd / 7 participants

Notations:

C: Credits filtering
I: Indoor/outdoor filtering
T: Shots threads filtering
R: Fast re-ranking

p1 = pers1 + T
p2 = pers2 + T
l1 = loc1 + C + I + R + T
l2 = loc2 + C + I + T

Θ: late fusion 1
Φ: late fusion 2
E: E conditions
A: A conditions

4 runs submitted:

F_E_IRIM1 = (p1 Θ p2) Θ (l1 Θ l2)
F_E_IRIM2 = p1 Θ (l1 Θ l2)
F_E_IRIM3 = p1 Θ l1
F_E_IRIM4 = p1 Φ l1

F_A_IRIM2 = p1 Θ (l1 Θ l2)
F_A_IRIM3 = p1 Θ l1
F_A_IRIM4 = p1 Φ l1

Runs

31 fully automatic runs submitted by 7 participants
6 first runs by PKU/ICST,   IRIM 2nd / 7 participants

Notations:

C: Credits filtering
I: Indoor/outdoor filtering
T: Shots threads filtering
R: Fast re-ranking

p1 = pers1 + T
p2 = pers2 + T
l1 = loc1 + C + I + R + T
l2 = loc2 + C + I + T

Θ: late fusion 1
Φ: late fusion 2
E: E condition
A: A condition

4 runs submitted:

F_E_IRIM1 = (p1 Θ p2) Θ (l1 Θ l2)
F_E_IRIM2 = p1 Θ (l1 Θ l2)
F_E_IRIM3 = p1 Θ l1
F_E_IRIM4 = p1 Φ l1

F_A_IRIM2 = p1 Θ (l1 Θ l2)
F_A_IRIM3 = p1 Θ l1
F_A_IRIM4 = p1 Φ l1

Rank	Run	mAP
1	F_E_PKU_ICST_1	0.5491
7	F_E_IRIM_1	0.4466
8	F_E_IRIM_2	0.4173
9	F_E_IRIM_3	0.4100
12	F_A_IRIM_2	0.3889
13	F_A_IRIM_3	0.3880
	Median run	0.3800
17	F_E_IRIM_4	0.3783
18	F_A_IRIM_4	0.3769

Analysis

- NIST provides « mixed-query » groundtruth

- Extraction of « person » and « location » from 2016 and 2017 queries.
  => incomplete groundtruth but it should give us an idea of methods performance

Analysis: Person recognition

Method	mAP 2016	mAP 2017
pers1A	0,1305	0,0613
pers1E	0,1425	0,0656
pers2E	0,1230	0,0448

Analysis: Person recognition

Method	mAP 2016	mAP 2017
pers1A	0,1305	0,0613
pers1A + T = p1A	0,1489	0,0708
pers1E	0,1425	0,0656
pers1E + T = p1E	0,1686	0,0769
pers2E	0,1230	0,0448
pers2E + T = p2E	0,1317	0,0484

Analysis: Person recognition

Method	mAP 2016	mAP 2017
pers1A	0,1305	0,0613
pers1A + T = p1A	0,1489	0,0708
pers1E	0,1425	0,0656
pers1E + T = p1E	0,1686	0,0769
pers2E	0,1230	0,0448
pers2E + T = p2E	0,1317	0,0484
p1E Θ p2E	0,1573	0,0827

Analysis: Location recognition

Loc1: Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

Analysis: Location recognition

<u>Loc1:</u> Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

<u>Re-ranking and filtering:</u>

Method	mAP 2016	mAP 2017
loc1E	0.2551	0.2075
loc2E	0.0663	0.0623

Analysis: Location recognition

Loc1: Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

Re-ranking and filtering:

Method	mAP 2016	mAP 2017
loc1E	0.2551	0.2075
loc1E + R	0.2965	0.2449
loc2E	0.0663	0.0623

Analysis: Location recognition

Loc1: Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

Re-ranking and filtering:

Method	mAP 2016	mAP 2017
loc1E	0.2551	0.2075
loc1E + R	0.2965	0.2449
loc1E + R + T	0.3292	0.2838
loc2E	0.0663	0.0623
loc2E + T	0.0999	0.0865

Analysis: Location recognition

Loc1: Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

Re-ranking and filtering:

Method	mAP 2016	mAP 2017
loc1E	0.2551	0.2075
loc1E + R	0.2965	0.2449
loc1E + R + T	0.3292	0.2838
loc1E + C + I + R + T = l1E	0.3302	0.2851
loc2E	0.0663	0.0623
loc2E + T	0.0999	0.0865
loc2E + C + I + T = l2E	0.1000	0.0863

Analysis: Location recognition

Loc1: Histogram normalization/distance

Method	mAP 2016	mAP 2017
loc1E (nL1/L1)	0.1836	0.1050
loc1E (nL2/L2)	0.1777	0.1334
loc1E (nL2/Cosine similarity)	0.2551	0.2075

Re-ranking and filtering:

Method	mAP 2016	mAP 2017
loc1E	0.2551	0.2075
loc1E + R	0.2965	0.2449
loc1E + R + T	0.3292	0.2838
loc1E + C + I + R + T = l1E	0.3302	0.2851
loc2E	0.0663	0.0623
loc2E + T	0.0999	0.0865
loc2E + C + I + T = l2E	0.1000	0.0863
l1E Θ l2E	0.3351	0.2862

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0{,}42$ $\alpha_P = 0{,}86$ $\alpha_L = 0{,}98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0,42$ $\alpha_P = 0,86$ $\alpha_L = 0,98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
p1 Θ (l1 Θ l2)	0.2954	0.4480

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0{,}42$ $\alpha_P = 0{,}86$ $\alpha_L = 0{,}98$ E condition

Run	**mAP 2016**	**mAP 2017**
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
p1 Θ (l1 Θ l2)	0.2954	0.4480
(p1 Θ p2) Θ l1	0.2919	0.4415

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0,42$ $\alpha_P = 0,86$ $\alpha_L = 0,98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
p1 Θ (l1 Θ l2)	0.2954	0.4480
(p1 Θ p2) Θ l1	0.2919	0.4415
p1 Θ l1	0.2874	0.4411

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0{,}42$ $\alpha_P = 0{,}86$ $\alpha_L = 0{,}98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
(p1 Θ p2) Θ ((loc1 + R + T) Θ (loc2 + T))	0.2984	0.4516
p1 Θ (l1 Θ l2)	0.2954	0.4480
(p1 Θ p2) Θ l1	0.2919	0.4415
p1 Θ l1	0.2874	0.4411

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0{,}42$ $\alpha_P = 0{,}86$ $\alpha_L = 0{,}98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
(p1 Θ p2) Θ ((loc1 + R + T) Θ (loc2 + T))	0.2984	0.4516
p1 Θ (l1 Θ l2)	0.2954	0.4480
p1 Θ ((loc1 + R + T) Θ (loc2 + T))	0.2949	0.4496
(p1 Θ p2) Θ l1	0.2919	0.4415
(p1 Θ p2) Θ (loc1 + R + T)	0.2907	0.4406
p1 Θ l1	0.2874	0.4411
p1 Θ (loc1 + R + T)	0.2858	0.4409

Analysis: Optimal runs

With optimal weights: $\alpha_G = 0{,}42$ $\alpha_P = 0{,}86$ $\alpha_L = 0{,}98$ E condition

Run	mAP 2016	mAP 2017
(p1 Θ p2) Θ (l1 Θ l2)	0.2984	0.4493
(p1 Θ p2) Θ ((loc1 + R + T) Θ (loc2 + T))	0.2984	0.4516
p1 Θ (l1 Θ l2)	0.2954	0.4480
p1 Θ ((loc1 + R + T) Θ (loc2 + T))	0.2949	0.4496
(p1 Θ p2) Θ l1	0.2919	0.4415
(p1 Θ p2) Θ (loc1 + R + T)	0.2907	0.4406
p1 Θ l1	0.2874	0.4411
p1 Θ (loc1 + R + T)	0.2858	0.4409

Conclusion

Fusion of heterogenous general methods

Conclusion

Fusion of heterogenous general methods

Significant progress from last year:

- Pers2 added
- Loc1 improved: L2 normalization/similarity and Re-ranking
- Shots Threads

Conclusion

Fusion of heterogenous general methods

Significant progress from last year:

- Pers2 added
- Loc1 improved: L2 normalization/similarity and Re-ranking
- Shots Threads

Future work:

- Improve Face track and Shots Threads
- Deeper understanding of the results
- Query expansion from Pers2 applied to Pers1 method

Thank you for your attention