Pixel retrieval

Content

- Pixel retrieval
- Benchmark and metrics
- Possible approaches
- Future directions

Pixel retrieval

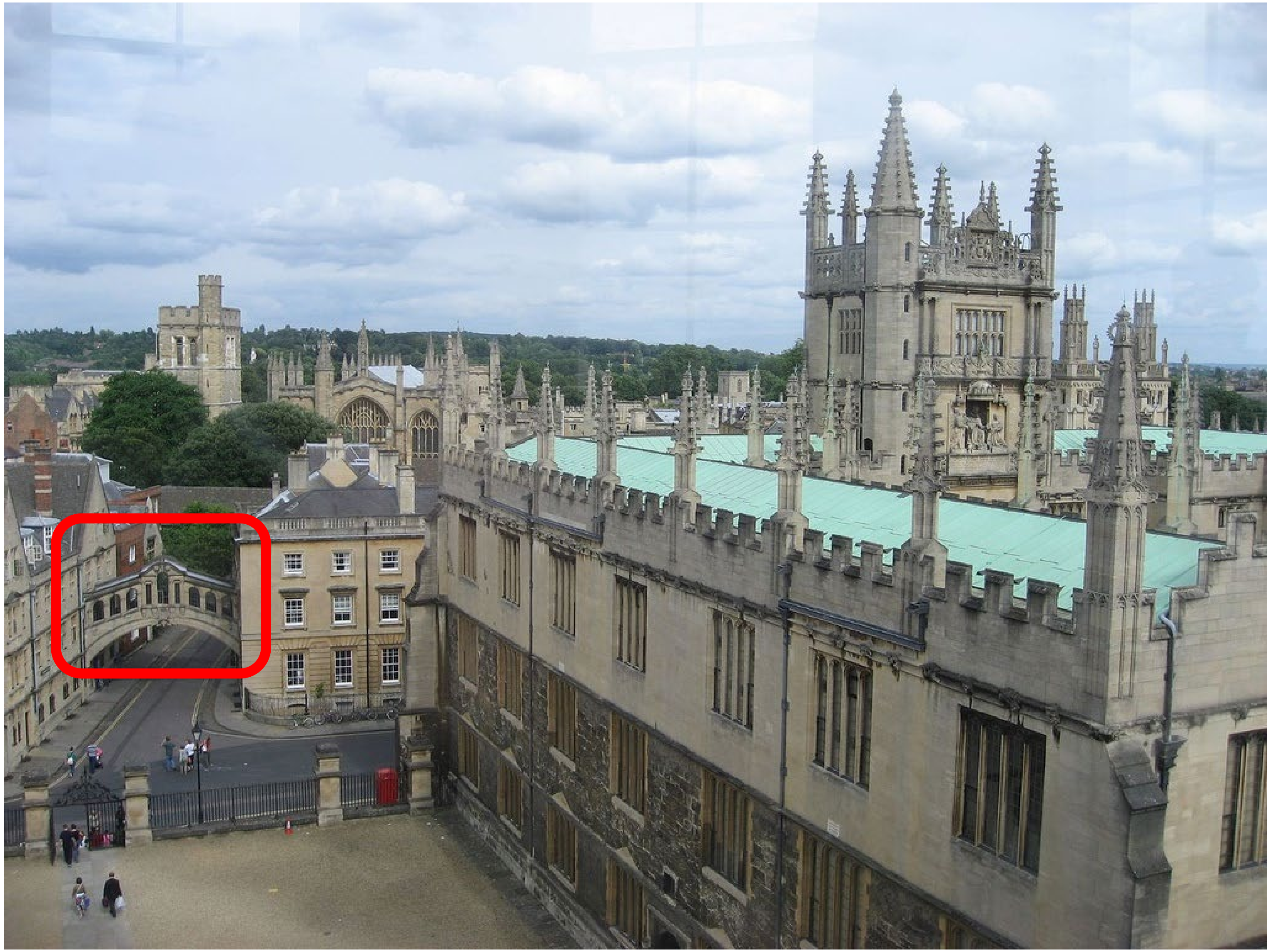
An issue of existing image retrieval

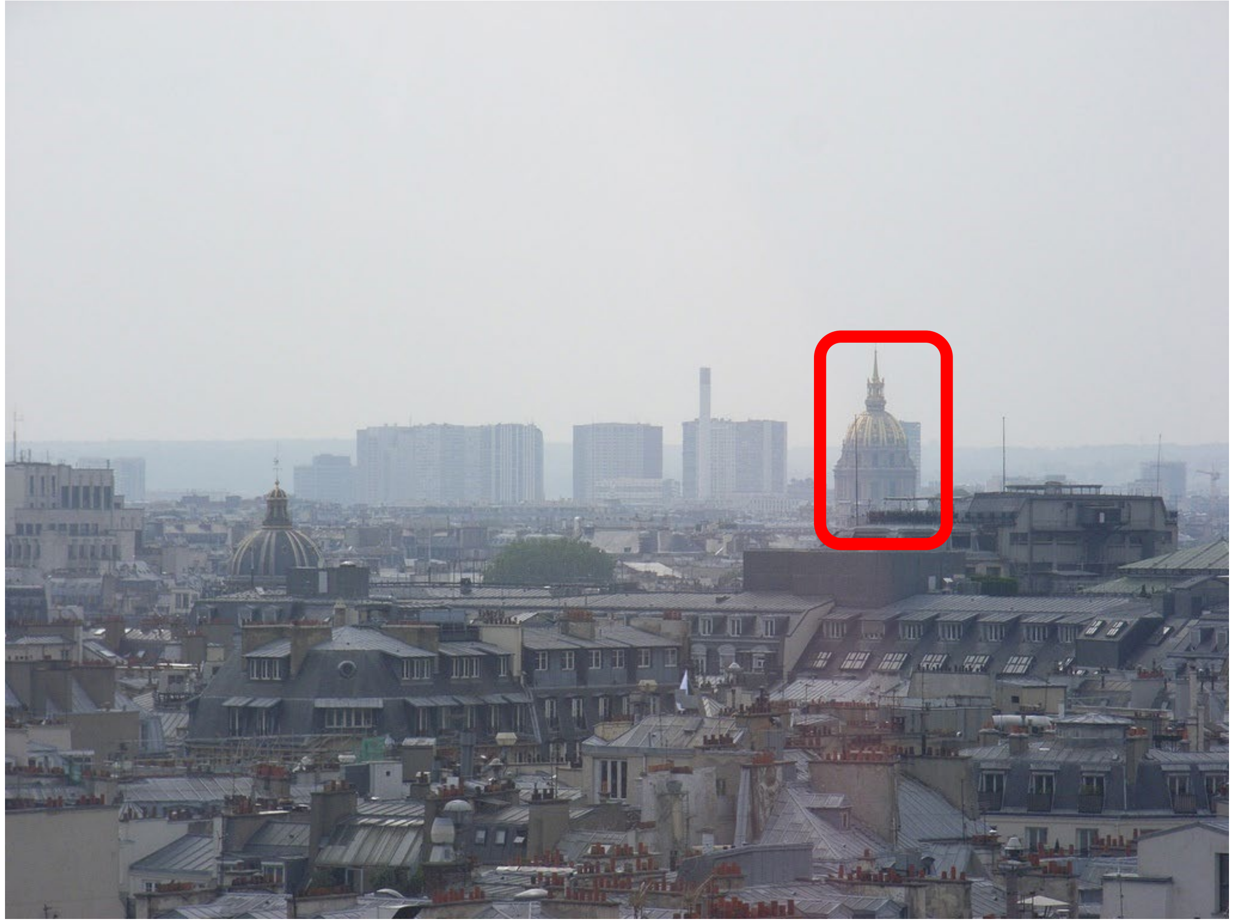
- Image retrieval
  - A real-world image has several different objects with complex background
  - Retrieved ranking list contains false positive images
  - Users may be difficult to identify the query object from the ranking list

Which image is correct?

Query

Image retrieval result





Pixel retrieval

- Image retrieval
  - Search the **images** which contain the query object from the database
- Pixel retrieval
  - Search **pixels** that depict the query object from the database
  - Retrieve, localize, and segment the target object from the database images

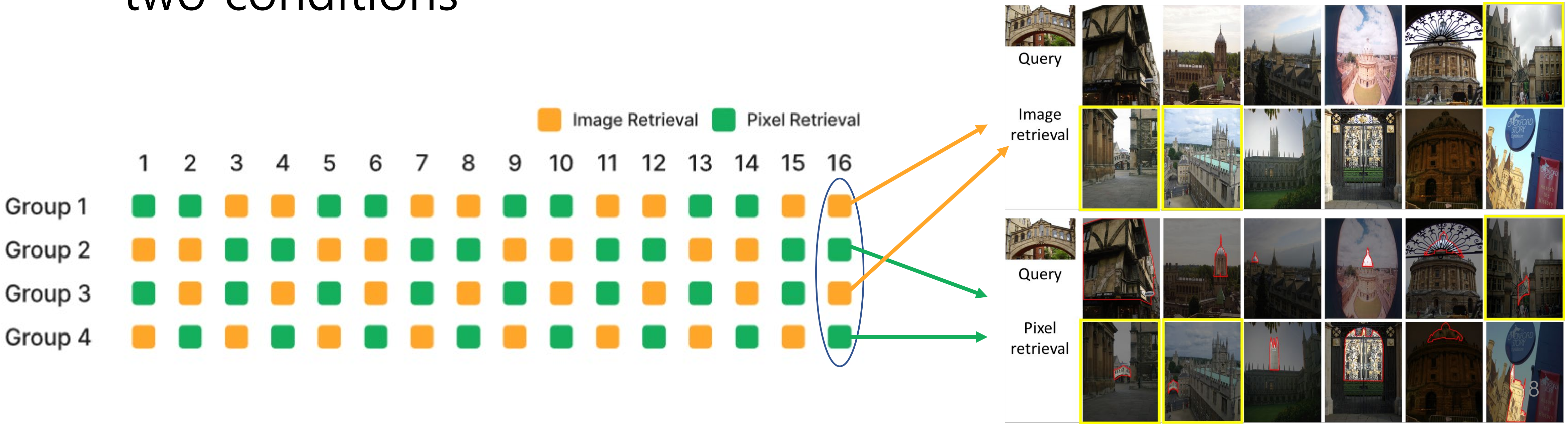
Which image is correct?

Query

Pixel retrieval result

An user study - setting

- 40 participants on Prolific divided into 4 groups
- 16 questions
  - Find images that contain a given target among candidate images
- Compare the **time** taken to complete the task between the two conditions



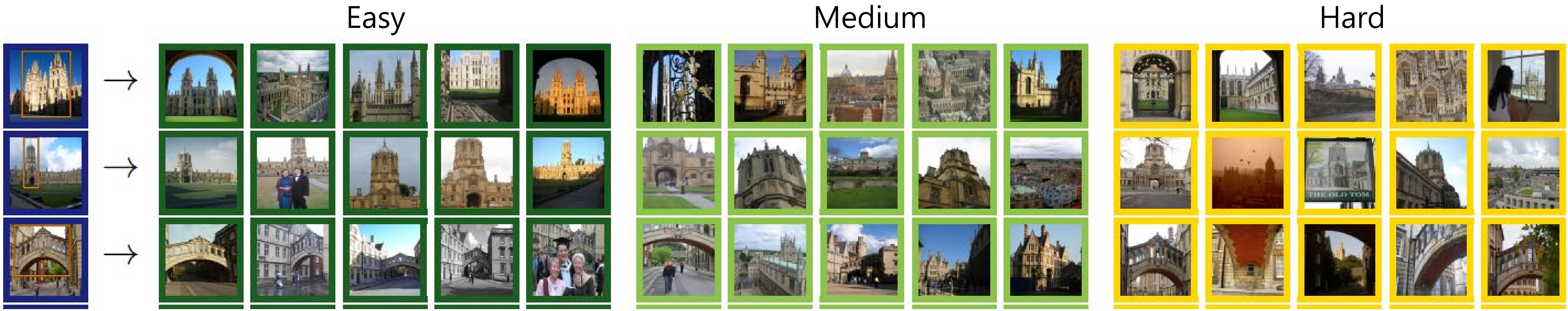
An user study - result

- Pixel retrieval help users to finish the task **faster**
  - Image retrieval: mean=53.71s, std=80.08s
  - Pixel retrieval: mean=37.07s, std=49.76s
- Difference is **statistically significant**
  - T-test, p-value=0.00091
- Participants responded that pixel retrieval annotations was **helpful**
  - Mean = 6.375/7, std = 0.89

Benchmarks and metrics

Benchmarks – data source

- Revisited Oxford and Paris
  - Introduced in 2007, 2008. Refined in 2018
  - 4996 images in Oxford, 6443 in Paris, and 1 million distractors
- Merit
  - 1: a **popular** benchmark in image retrieval
  - 2: **severe** viewpoint changes, occlusions, and illumination changes
  - 3: each query image contains up to **hundreds of positive database images**, while other datasets, such as UKBench [27] and Holiday [12], only have 4 to 5 positive images for each query

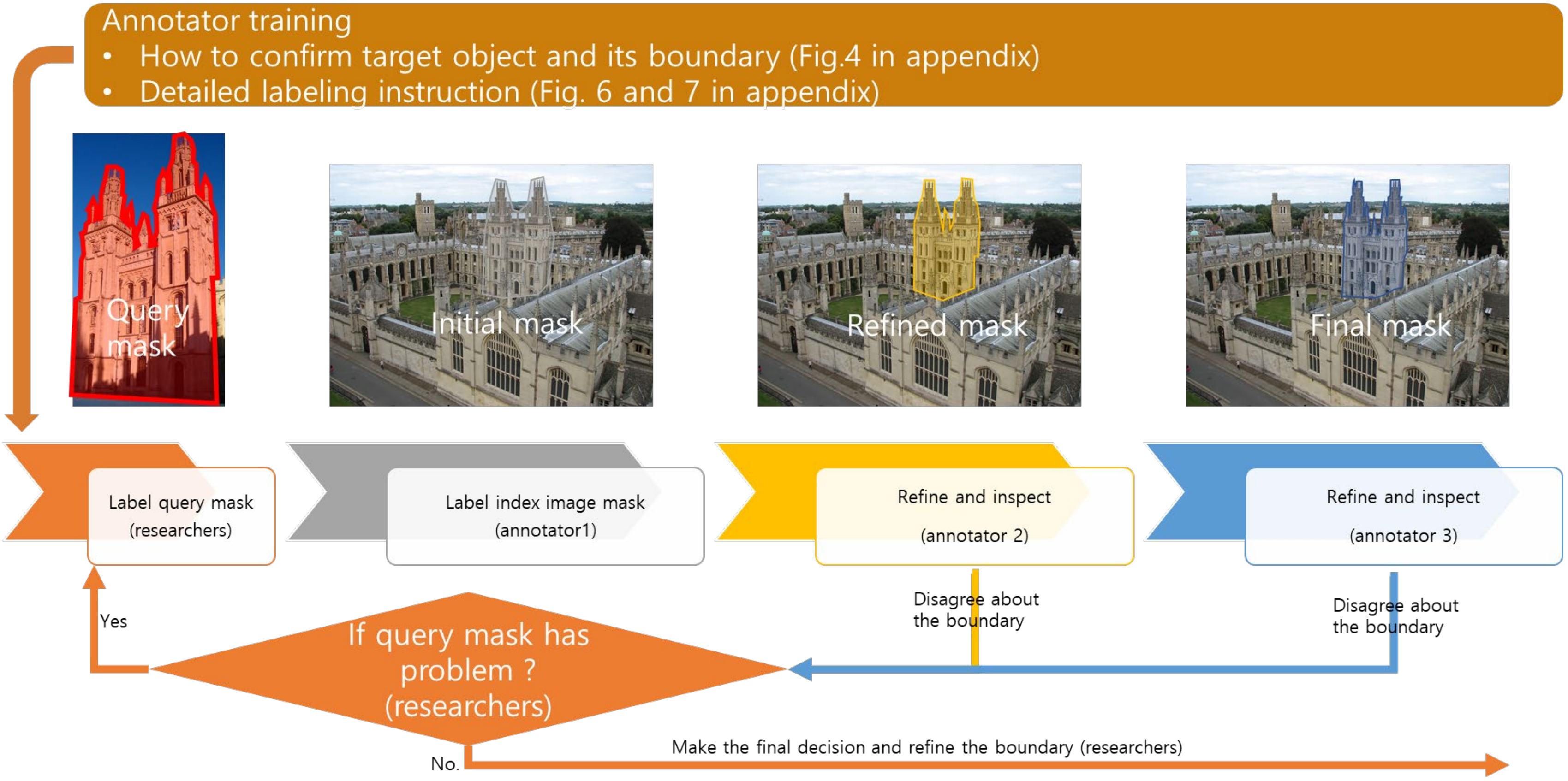


Query

up to hundreds of positive database images

Benchmark - Annotation

- Mask annotation
  - Query: researchers
  - DB images: annotators
- Quality assurance
  - 3 professional annotators
  - 3 steps



Benchmark - Metrics

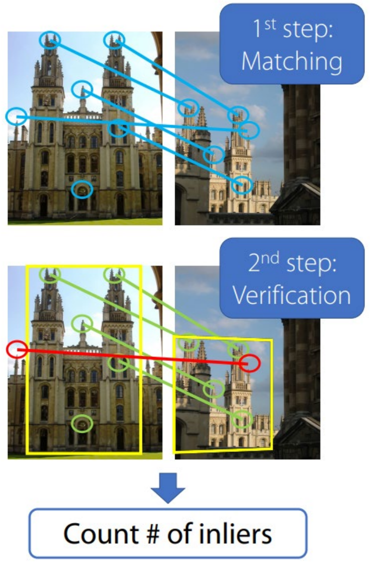
- Pixel retrieval from database
  - Existing image retrieval metric: **mAP**
  - New pixel retrieval metric: **mAP@50:5:95**
    - An database image is true positive only if its **Intersection over Union (IoU)** is larger than a threshold **n**
    - **n** is set from 0.5 to 0.95, with step 0.05
- Pixel retrieval from ground-truth query-index image pairs
  - Use existing ranking/reranking methods and treat the remaining process as one-shot detection/segmentation
  - Metric: **mean of mIoU** of all the queries, where mIoU is the mean of the IoUs for all the ground-truth index images

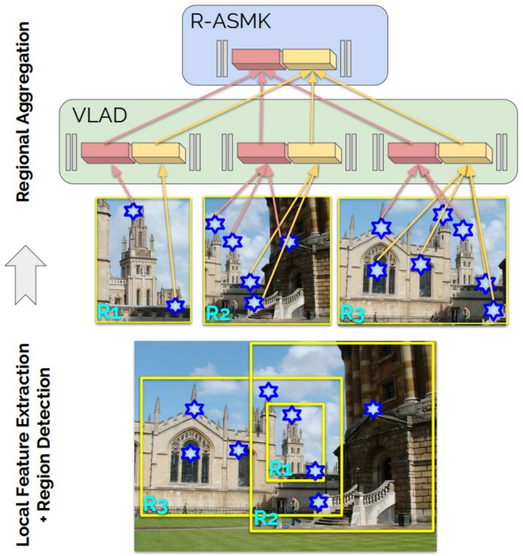
Possible approaches

1. Retrieve the images from database, which is image retrieval.
2. Detect and segment the target object from the retrieved images.

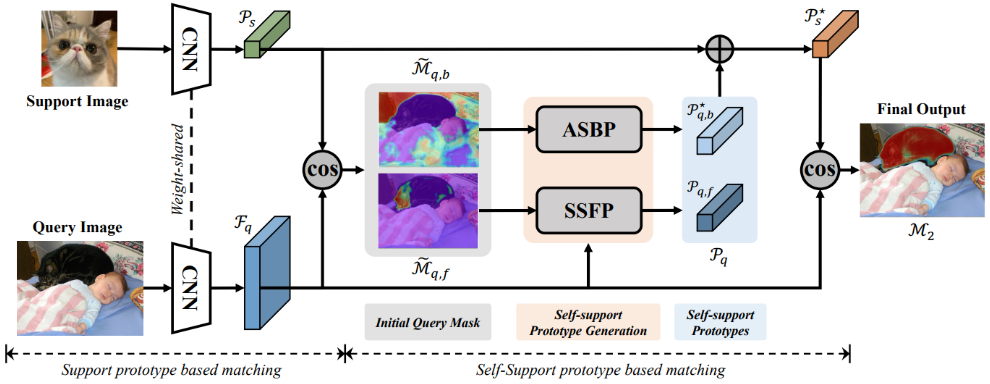
Existing approaches

- Retrieval methods
  - Spatial verification
  - Detect-2-retrieval



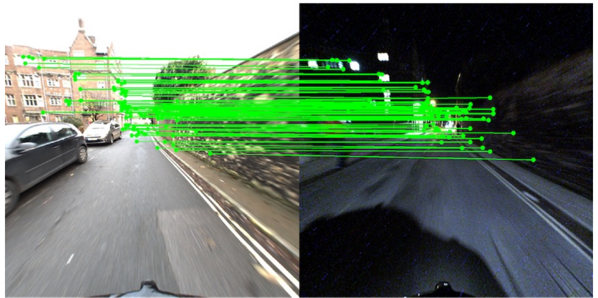


- One-shot detection and segmentation
  - Open world localization
  - HSNet, SSP, …



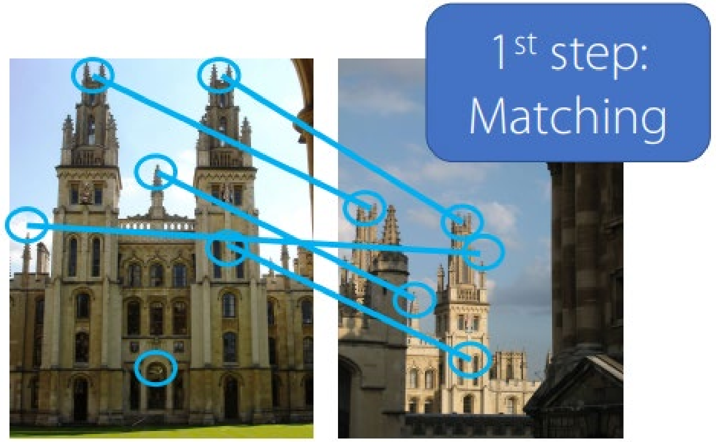
- Dense matching
  - GLUNet, WarpC, …

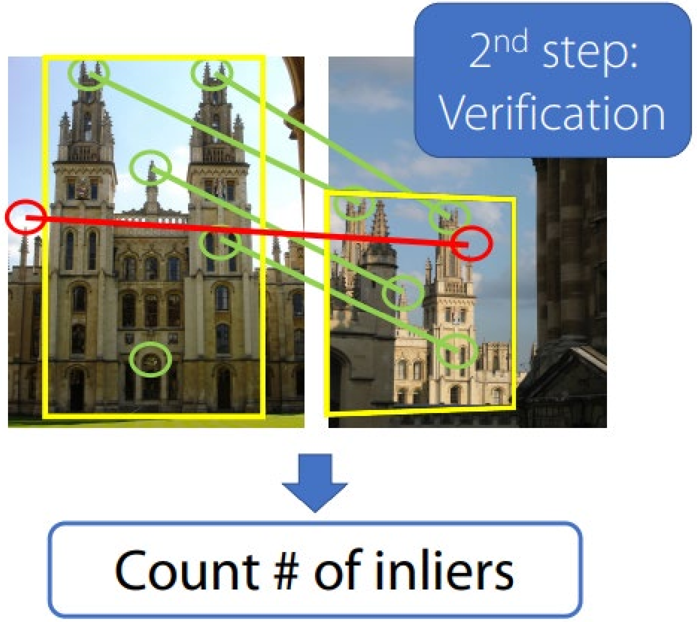
However, they have to **combined with retrieval methods** to achieve pixel retrieval.



Spatial verification

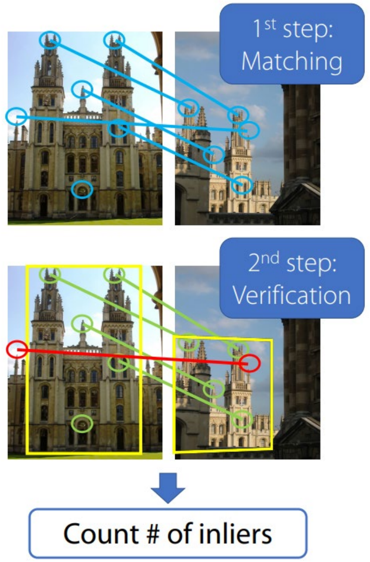
- Detect **keypoints** from two images, and **match** them
- The matched keypoints have both outliers and inliers
- Use spatial model to verify them

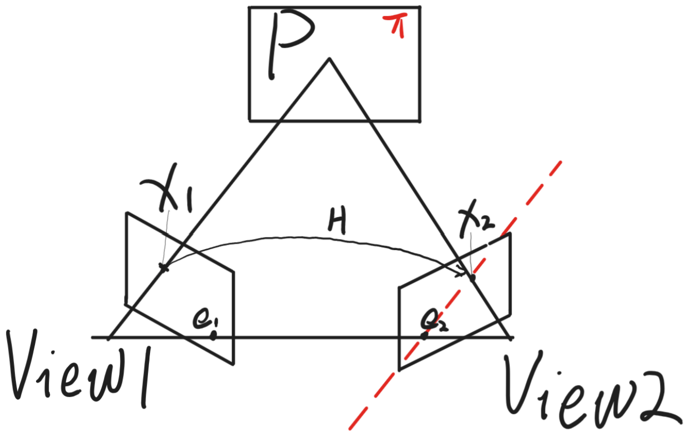




Spatial verification (Cont.)

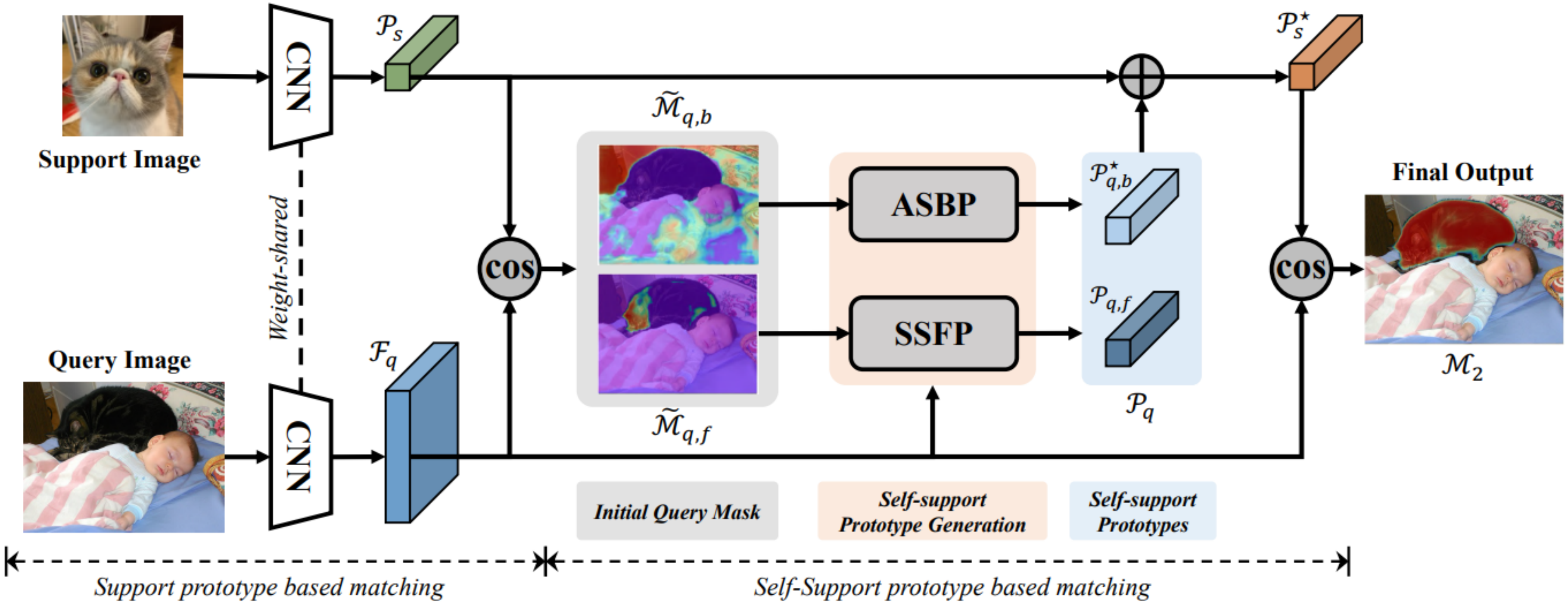
- Matching in 1st step has inliers and outliers
- Can use a homography matrix (H) to describe the spatial configuration change between point locations in different views
  - $x_2 = x_1 * H$
- But we do not know the H
  - Solution: repeatedly sample H, and select the one with the highest number of inliers.
  - If $x_1 H - x_2 < \varepsilon$, the matching for x_1 *and* x_2 is a inlier
  - This process is called random sample consensus (RANSAC)





One-shot detection and segmentation

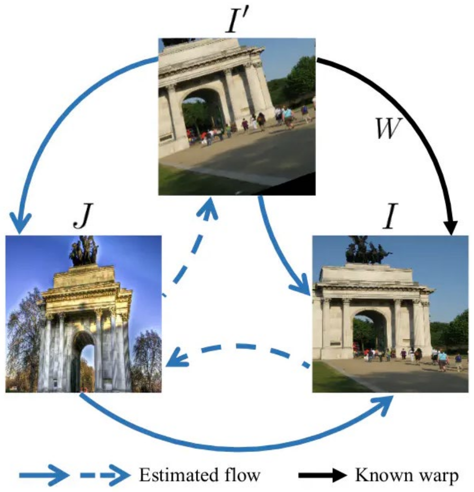
- Self-support few-shot segmentation (SSP)

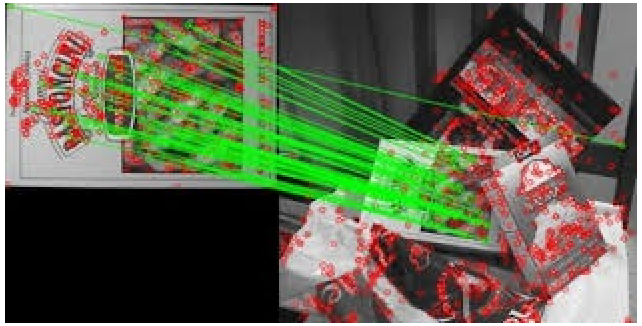


Qi Fan,... Self-support few-shot semantic segmentation, ECCV 2022

Dense matching

- Warp consistency for unsupervised training

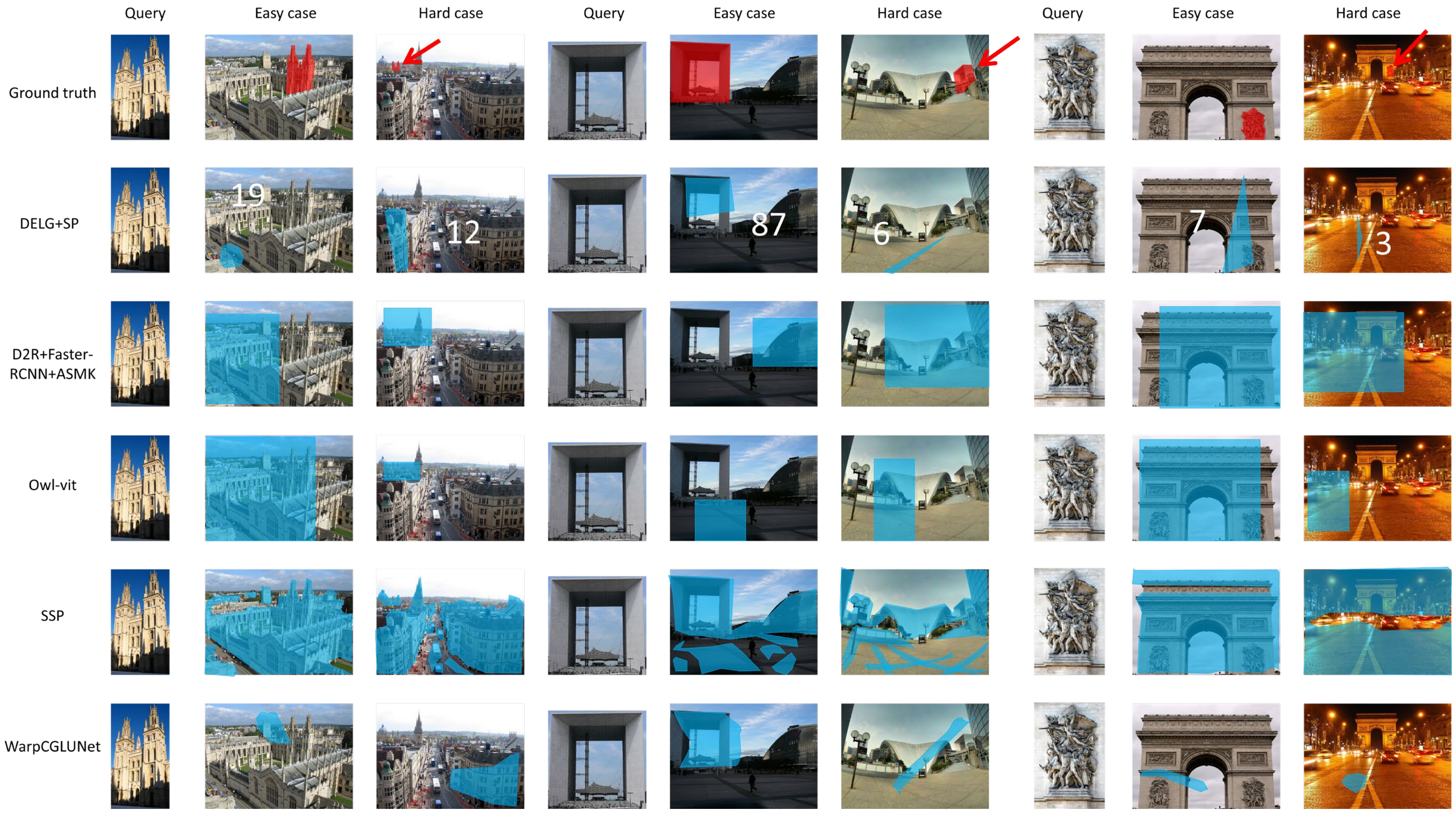


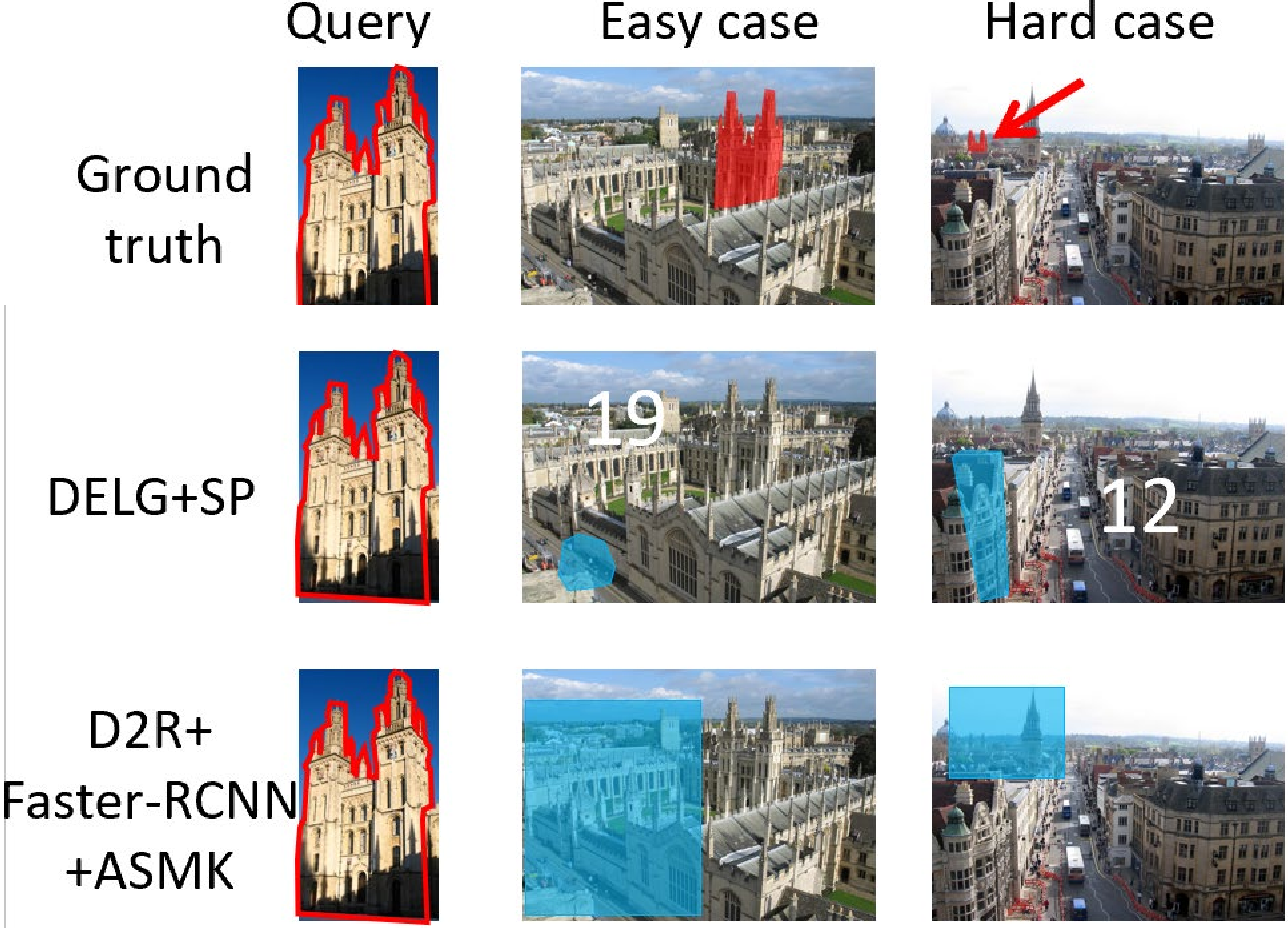


Truong,… Warp Consistency for Unsupervised Learning of Dense Correspondences, ICCV 2021

Qualitative results

- unsatisfied performance on hard cases.



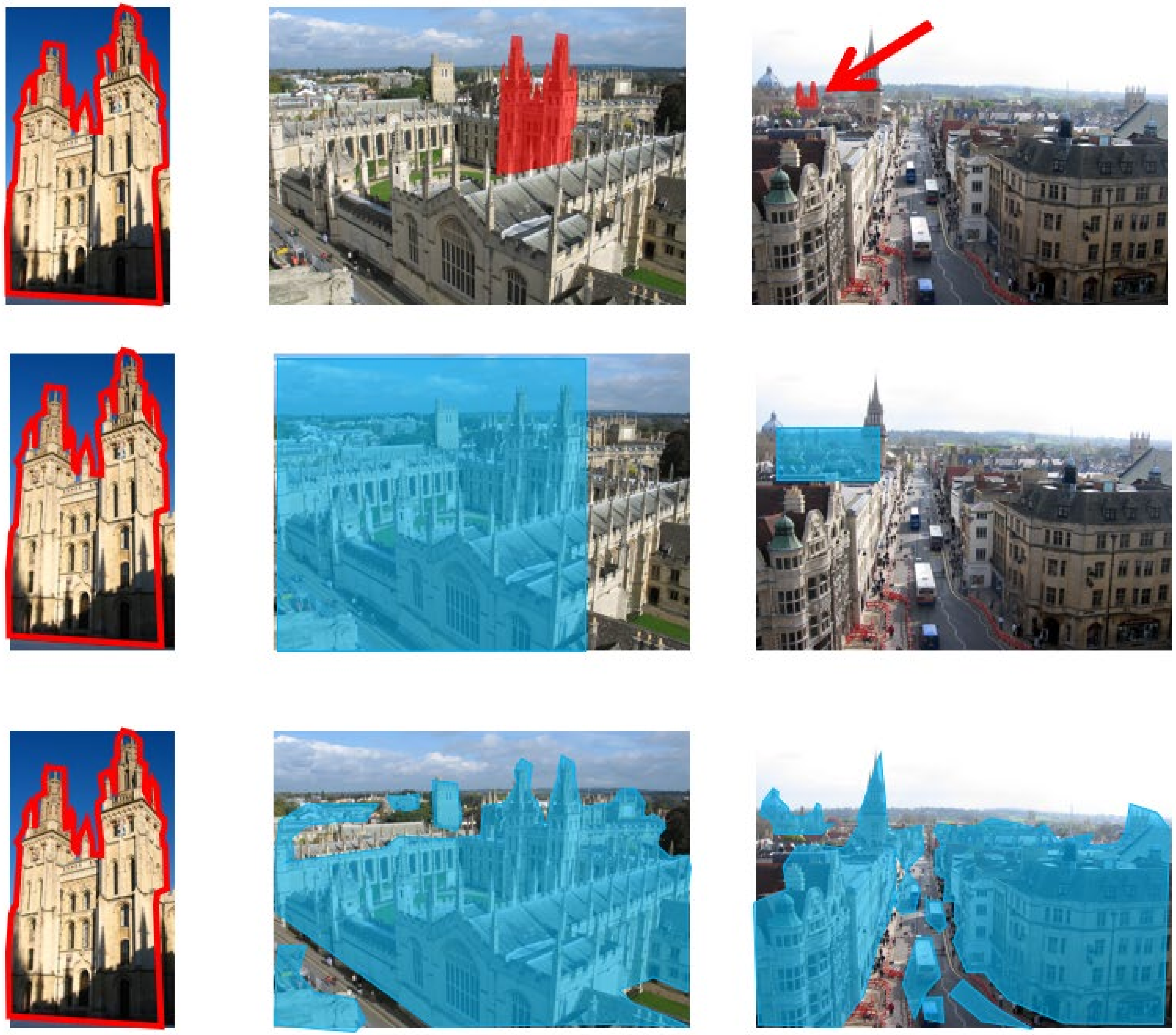

Query
Easy case
Hard case
Ground truth
DELG+SP
19
12
D2R+ Faster-RCNN +ASMK

Query | Easy case | Hard case

Ground truth

Owl-vit

SSP



Testbed

- No single method outperforms all others across all test protocols
- Methods with high accuracy are usually slow

Method	Medium				Hard				Average
	PROxf		PRPar		PROxf		PRPar		
	D	S	D	S	D	S	D	S	
Retrieval and localization unified methods									
SIFT+SP [27]	26.1	10.9	24.2	9.7	18.2	7.3	19.3	7.8	15.44
DELF+SP [24]	43.7	20.0	40.7	16.7	33.2	13.9	32.2	12.4	26.60
DELG+SP [4]	**44.1**	19.7	40.1	16.5	**34.8**	14.5	31.2	11.7	26.57
D2R [35]+Resnet-50-Faster-RCNN+Mean	20.2	-	29.6	-	16.7	-	27.4	-	-
D2R [35]+Resnet-50-Faster-RCNN+VLAD [16]	25.8	-	37.5	-	21.6	-	35.5	-	-
D2R [35]+Resnet-50-Faster-RCNN+ASMK [36]	26.3	-	38.5	-	21.6	-	35.6	-	-
D2R [35]+Mobilenet-V2-SSD+Mean	19.7	-	25.9	-	20.1	-	27.9	-	-
D2R [35]+Mobilenet-V2-SSD+VLAD [16]	23.1	-	33.	-	20.9	-	33.6	-	-
D2R [35]+Mobilenet-V2-SSD+ASMK [36]	22.4	-	34.0	-	20.8	-	33.1	-	-
Detection methods									
OWL-VIT (LiT) [22]	11.4	-	18.0	-	6.3	-	15.0	-	-
OS2D-v2-trained [25]	10.5	-	13.7	-	11.7	-	14.3	-	-
OS2D-v1 [25]	7.0	-	8.5	-	8.7	-	9.2	-	-
OS2D-v2-init [25]	13.6	-	15.4	-	14.0	-	15.1	-	-
Segmentation methods									
SSP (COCO) + ResNet50 [11]	19.2	34.5	31.1	48.7	15.1	25.3	29.8	**41.7**	30.68
SSP (VOC) + ResNet50 [11]	19.7	34.3	31.4	**48.8**	16.1	**26.1**	30.3	40.4	30.89
HSNet (COCO) + ResNet50 [21]	23.4	32.8	37.4	41.9	21.0	25.7	34.7	36.5	31.67
HSNet (VOC) + ResNet50 [21]	21.0	29.8	31.4	39.7	17.1	23.2	29.7	34.9	28.35
HSNet (FSS) + ResNet50 [21]	30.5	**35.7**	39.4	40.2	22.7	25.1	34.7	32.8	**32.64**
Mining (VOC) + ResNet50 [46]	18.3	30.5	29.6	42.7	15.1	21.4	28.1	34.3	27.50
Mining (VOC) + ResNet101 [46]	18.1	28.6	29.5	40.0	14.2	20.4	28.2	34.4	26.68
Dense matching methods									
GLUNet-Geometric [39]	18.1	13.2	22.8	15.2	7.7	4.6	13.3	7.8	12.84
PDCNet-Geometric [40]	29.1	24.0	30.7	21.9	20.4	15.7	20.6	12.6	21.87
GOCor-GLUNet-Geometric [38]	30.4	26.0	33.4	25.6	20.8	16.0	19.8	13.3	23.16
WarpC-GLUNet-Geometric (megadepth) [41]	31.3	25.4	36.6	27.3	21.9	15.8	26.4	17.3	25.25
GLUNet-Semantic [39]	18.5	14.4	22.4	15.6	8.7	5.6	12.8	7.8	13.22
WarpC-GLUNet-Semantic [41]	27.5	21.4	36.8	25.7	18.5	11.9	28.3	17.6	23.46

Pixel retrieval from ground-truth query-index image pairs

		PROxf		PROxf+R1M		PRPar		PRPar+R1M		Overhead per 100 image pairs
		M	H	M	H	M	H	M	H	
Image retrieval: DELG initial ranking [4] + HD reranking [1]										
Pixel retrieval methods	DELG + SP [4]	39.6	30.5	36.0	28.2	34.8	20.2	34.7	19.5	41.22s
	D2R+Faster-RCNN+ASMK [35]	30.1	23.5	30.5	22.0	26.3	25.3	25.7	24.9	0.11 s
	OWL-VIT [22]	12.3	6.6	12.1	13.6	7.9	7.6	7.9	7.8	296.21s
	SSP [11]	**33.0**	29.7	**35.7**	30.5	46.4	**37.2**	45.6	**37.2**	62.33 s
	WarpCGLUNet [41]	31.2	**32.6**	31.5	**31.7**	34.1	27.3	34.3	28.1	181.64s

Pixel retrieval from database

- No single method outperforms all others across all test protocols

Method	Medium				Hard				Average
	PROxf		PRPar		PROxf		PRPar		
	D	S	D	S	D	S	D	S	
Retrieval and localization unified methods									
SIFT+SP [27]	26.1	10.9	24.2	9.7	18.2	7.3	19.3	7.8	15.44
DELF+SP [24]	43.7	20.0	40.7	16.7	33.2	13.9	32.2	12.4	26.60
DELG+SP [4]	**44.1**	19.7	40.1	16.5	**34.8**	14.5	31.2	11.7	26.57
D2R [35]+Resnet-50-Faster-RCNN+Mean	20.2	-	29.6	-	16.7	-	27.4	-	-
D2R [35]+Resnet-50-Faster-RCNN+VLAD [16]	25.8	-	37.5	-	21.6	-	35.5	-	-
D2R [35]+Resnet-50-Faster-RCNN+ASMK [36]	26.3	-	38.5	-	21.6	-	35.6	-	-
D2R [35]+Mobilenet-V2-SSD+Mean	19.7	-	25.9	-	20.1	-	27.9	-	-
D2R [35]+Mobilenet-V2-SSD+VLAD [16]	23.1	-	33.	-	20.9	-	33.6	-	-
D2R [35]+Mobilenet-V2-SSD+ASMK [36]	22.4	-	34.0	-	20.8	-	33.1	-	-
Detection methods									
OWL-VIT (LiT) [22]	11.4	-	18.0	-	6.3	-	15.0	-	-
OS2D-v2-trained [25]	10.5	-	13.7	-	11.7	-	14.3	-	-
OS2D-v1 [25]	7.0	-	8.5	-	8.7	-	9.2	-	-
OS2D-v2-init [25]	13.6	-	15.4	-	14.0	-	15.1	-	-
Segmentation methods									
SSP (COCO) + ResNet50 [11]	19.2	34.5	31.1	48.7	15.1	25.3	29.8	**41.7**	30.68
SSP (VOC) + ResNet50 [11]	19.7	34.3	31.4	**48.8**	16.1	**26.1**	30.3	40.4	30.89
HSNet (COCO) + ResNet50 [21]	23.4	32.8	37.4	41.9	21.0	25.7	34.7	36.5	31.67
HSNet (VOC) + ResNet50 [21]	21.0	29.8	31.4	39.7	17.1	23.2	29.7	34.9	28.35
HSNet (FSS) + ResNet50 [21]	30.5	**35.7**	39.4	40.2	22.7	25.1	34.7	32.8	**32.64**
Mining (VOC) + ResNet50 [46]	18.3	30.5	29.6	42.7	15.1	21.4	28.1	34.3	27.50
Mining (VOC) + ResNet101 [46]	18.1	28.6	29.5	40.0	14.2	20.4	28.2	34.4	26.68
Dense matching methods									
GLUNet-Geometric [39]	18.1	13.2	22.8	15.2	7.7	4.6	13.3	7.8	12.84
PDCNet-Geometric [40]	29.1	24.0	30.7	21.9	20.4	15.7	20.6	12.6	21.87
GOCor-GLUNet-Geometric [38]	30.4	26.0	33.4	25.6	20.8	16.0	19.8	13.3	23.16
WarpC-GLUNet-Geometric (megadepth) [41]	31.3	25.4	36.6	27.3	21.9	15.8	26.4	17.3	25.25
GLUNet-Semantic [39]	18.5	14.4	22.4	15.6	8.7	5.6	12.8	7.8	13.22
WarpC-GLUNet-Semantic [41]	27.5	21.4	36.8	25.7	18.5	11.9	28.3	17.6	23.46

Pixel retrieval from ground-truth query-index image pairs

- Methods with high accuracy are slow

		PROxf		PROxf+R1M		PRPar		PRPar+R1M		Overhead per 100 image pairs
		M	H	M	H	M	H	M	H	
Image retrieval: DELG initial ranking [4] + HD reranking [1]										
Pixel retrieval methods	DELG + SP [4]	39.6	30.5	36.0	28.2	34.8	20.2	34.7	19.5	41.22s
	D2R+Faster-RCNN+ASMK [35]	30.1	23.5	30.5	22.0	26.3	25.3	25.7	24.9	0.11 s
	OWL-VIT [22]	12.3	6.6	12.1	13.6	7.9	7.6	7.9	7.8	296.21s
	SSP [11]	**33.0**	29.7	**35.7**	30.5	**46.4**	**37.2**	**45.6**	**37.2**	62.33 s
	WarpCGLUNet [41]	31.2	**32.6**	31.5	**31.7**	34.1	27.3	34.3	28.1	181.64s

Pixel retrieval <u>from database</u>

The objective should be to develop a method that
achieves both high accuracy and rapid processing speed.

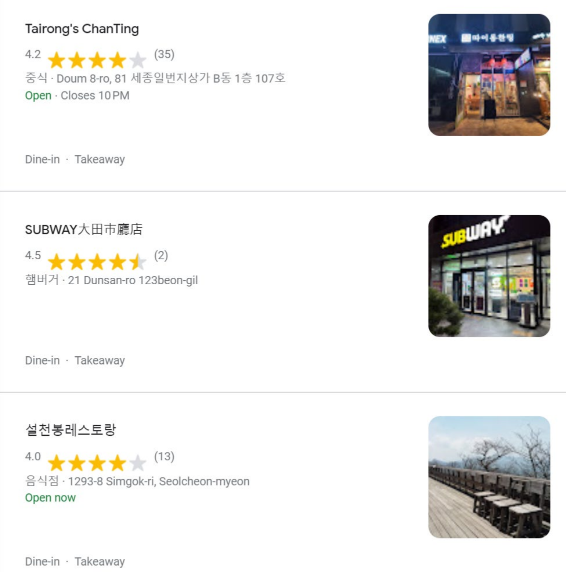
Future directions

Future directions

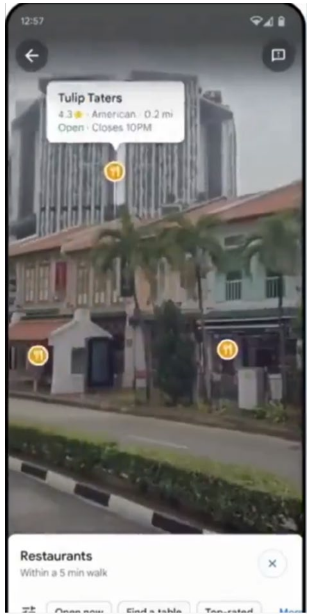
- Accuracy
- Speed
- Applications

Application 1: lens in map

- Detect and recognize shops.
- Lens in map



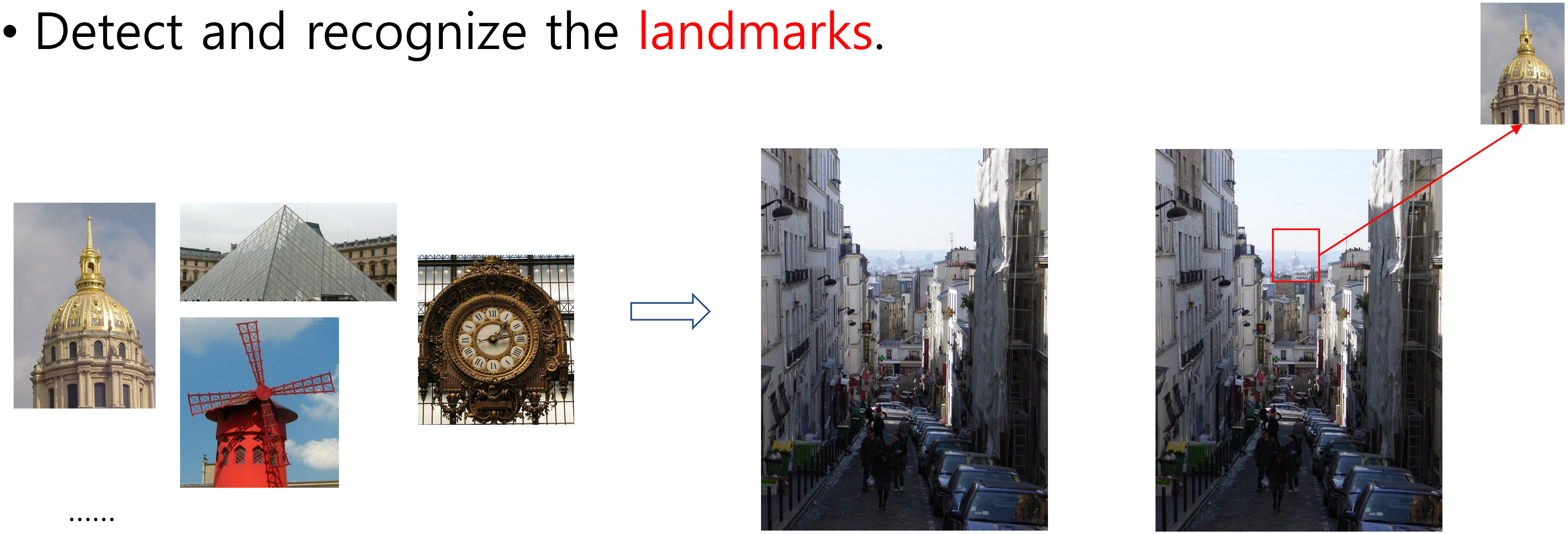
Given shop photos



Detect and recognize

Application 2: digital tour guide in city

- Detect and recognize the landmarks.



Given target objects Test image Detect and recognize

Application 3: digital tour guide in details

- Detect and recognize the landmark details.

二龙戏蜘蛛：不同于常见的二龙戏珠，这是全国唯一一个二龙戏蜘蛛的雕像。蜘蛛象征着商人希望人脉和生意像蜘蛛丝一样连接。"Two Dragons Playing with a Spider: Unlike the common motif of two dragons playing with a pearl, this is the only statue in the country of two dragons playing with a spider (pearl and spider share same pronunciation in Chinese). The spider symbolizes the merchant's hope that connections and business will be interlinked like spider webs."

宝剑和花瓶，谐音寓意保平安。"Treasured sword and vase, a homophonic expression symbolizing the assurance of peace and safety."

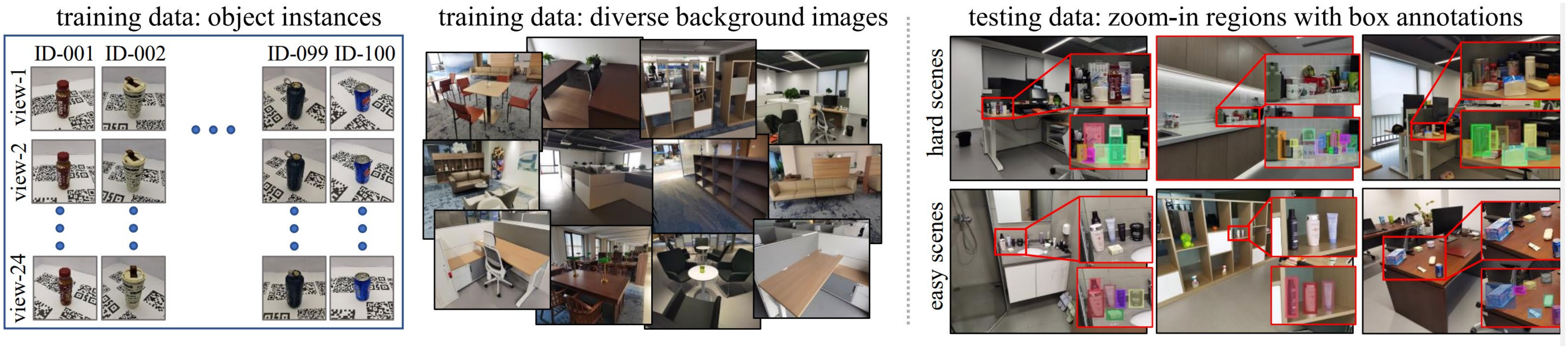
上面的是记账本，下面的是出账本。记账本打开，出账本合住，意味着只进不出，积财积富。"The one on top is the ledger for income, the one below is the ledger for expenses. The income ledger is open, the expense ledger is closed, signifying money only comes in and does not go out, accumulating wealth and riches."

Given target objects

Detect and recognize

Application 4: indoor robotics

- Detect and recognize the indoor instances.



[A High-Resolution Dataset for Instance Detection with Multi-View Instance Capture, NeurIPS 24]()

Q&A