Supplementary material for "On Train-Test Class Overlap and Detection for Image Retrieval"

Chull Hwan Song[1] Jooyoung Yoon[1] Taebaek Hwang[1] Shunghyun Choi[1]
Yeong Hyeon Gu[2†] Yannis Avrithis[3]
[1]Dealicious Inc. [2]Sejong University [3]Institute of Advanced Research on Artificial Intelligence (IARAI)

A. Implementation details

In our experiments, we use a computational environment featuring 8 RTX 3090 GPUs with PyTorch [29]. We perform transfer learning from models pre-trained on ImageNet [42]. To ensure a fair comparison with previous studies [59, 46, 58], we configure the learning environment as closely as possible. Specifically, we use ResNet101 [62] as the backbone with final feature dimension $d = 2048$.

We use ArcFace [6] loss function for training, with margin parameter 0.3. For optimization, we use stochastic gradient descent with momentum 0.9, weight decay 0.00001, initial learning rate 0.001, a warm-up phase [11] of three epochs and cosine annealing. We train SfM-120k for 100 epochs and $\mathcal{R}$GLDv2-clean for 50 epochs. Previous work has shown the effectiveness of preserving the original image resolution during the training of image retrieval models [10, 8]. We adopt this principle following [60, 46, 47], where each training batch consists of images with similar aspect ratios instead of a single fixed size. The batch size is 128. Following DIR [8] and DELF [28], we carry out classification-based training of the backbone only and subsequently fine-tune the model. During fine-tuning, we train CiDeR while the backbone is frozen, as shown in Figure A10.

For evaluation, we use multi-resolution representation [8] on both query and database images, applying ℓ_2-normalization and whitening [36] on the final features.

NETWORK	#PARAMS (M)	#GFLOPS
R101	42.50	7.86
Yokoo *et al.* [60]	43.91	7.86
SOLAR [27]	53.36	8.57
DOLG [59]	47.07	8.07
Token [58]	54.43	8.05
CiDeR (Ours)	46.12	7.94

Table A10. *Model complexity*: Parameters (#PARAMS) and computational complexity (#GFLOPS) of different models providing official code. Single forward pass, given an input image of size 224 × 224.

†Corresponding author

R101	BE	SC	AL	POOLING	#PARAMS (M)	#GFLOPS
✓					42.50	7.86
✓	✓				43.58	7.91
✓	✓	✓			43.88	7.93
✓	✓	✓	✓		44.02	7.94
✓	✓	✓	✓	✓	46.12	7.94

Table A11. *Model complexity*: Parameters (#PARAMS) and computational complexity (#GFLOPS) for different components of CiDeR. BE: backbone enhancement; SC: selective context; AL: attentional localization; Pooling: spatial pooling (GeM) + FC.

B. Model complexity

Table A10 compares the model complexity[1] of CiDeR with other models. In this table, R101 is the baseline for all related studies, all of which use the feature maps of its last layer. We observe that our model has the least complexity after Yokoo *et al.* [60], which only uses GeM + FC. Table A11 shows model complexity for each of the components of CiDeR, as defined in subsection 5.1.

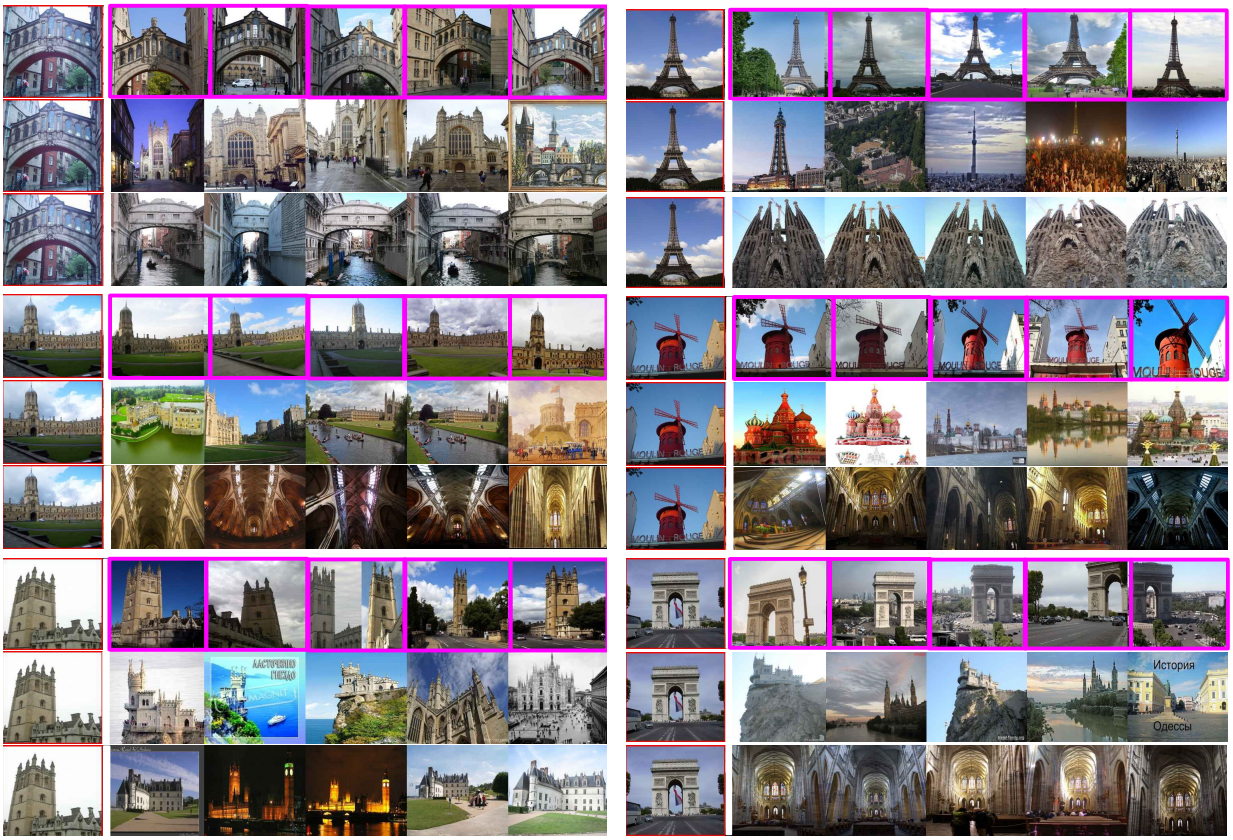
C. More on revisited *vs.* original GLDv2-clean

Details To identify overlapping landmarks, we use GLAM [46] to extract image features from the training and evaluation sets. Extracted features from the training sets are indexed using the Approximate Nearest Neighbor (ANN)[2] search method. For verification, we use SIFT [23] local descriptors. We find tentative correspondences between local descriptors by a kd-tree and we verify by obtaining inlier correspondences using RANSAC.

In addition to Figure 2 in section 3, Figure A8 shows overlapping landmark categories between the training set (GLDv2-clean, NC-clean, SfM-120k) and the evaluation set ($\mathcal{R}$Oxford, $\mathcal{R}$Paris). Clearly, only GLDv2-clean has overlapping categories with the evaluation set.

Table A12 shows the details of the 18 GIDs that are removed from GLDv2-clean due to overlap with the evaluation

[1] https://github.com/sovrasov/flops-counter.pytorch
[2] https://github.com/kakao/n2



Evaluation: $\mathcal{R}$Oxford Evaluation: $\mathcal{R}$Paris

Figure A8. *Confirming overlapping landmark categories* between training sets and evaluation sets ($\mathcal{R}$Oxford, $\mathcal{R}$Paris). Red box: query image. The query image from the evaluation set in each box/row is followed by top-5 most similar images from the training set (for each query, top down: GLDv2-clean, NC-clean, SfM-120k). Pink box: training image landmark identical with query (evaluation) image landmark.

#	GID	# Images	GLDv2 Landmark Name	Oxford/Paris Landmark Name
1	6190	98	Radcliffe Camera	Oxford
2	19172	32	All Souls College, Oxford	All Souls Oxford
3	37135	18	Oxford University Museum of Natural History	Pitt Rivers Oxford
4	42489	55	Pont au Double	Jesus Oxford
5	147275	18	Magdalen Tower	Ashmolean Oxford
6	152496	71	Christ Church, Oxford	Christ Church Oxford
7	167275	55	Bridge of Sighs (Oxford)	Magdalen Oxford
8	181291	60	Petit-Pont	Notre Dame Paris
9	192090	23	Christ Church Great Quadrangle	Paris
10	28949	91	Moulin Rouge	Moulin Rouge Paris
11	44923	41	Jardin de l'Intendant	Hotel des Invalides Paris
12	47378	731	Eiffel Tower	Eiffel Tower Paris
13	69195	34	Place Charles-de-Gaulle (Paris)	Arc de Triomphe Paris
14	167104	23	Hôtel des Invalides	Hotel des Invalides Paris
15	145268	72	Louvre Pyramid	Louvre Paris
16	146388	80	Basilique du Sacré-Cœur de Montmartre	Arc de Triomphe Paris
17	138332	30	Parvis Notre-Dame - place Jean-Paul-II (Paris)	Notre Dame Paris
18	144472	33	Esplanade des Invalides	Paris

Table A12. Details of GIDs removed from GLDv2-clean dataset.

sets. The new, revisited $\mathcal{R}$GLDv2-clean dataset is what remains after this removal.

Classes with/without overlap Table A13 elaborates on the results of Table 4 by comparing the original GLDv2-clean training set with our revisited version $\mathcal{R}$GLDv2-clean

Method	trainset	OC	Ox5k	Par6k	Medium		Hard		Mean
					$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par	
SOLAR [58]	GLDv2-clean	Y	82.1	95.2	72.5	88.8	47.3	75.8	77.0
		N	81.6	95.9	66.1	84.8	44.3	70.6	73.9
SOLAR [27]†	$\mathcal{R}$GLDv2-clean	Y	77.7	87.6	65.4	78.4	36.0	62.2	67.9
		N	80.1	92.0	66.3	81.6	42.6	68.7	71.9
GLAM [46]	GLDv2-clean	Y	81.6	93.9	73.6	88.6	53.6	77.4	78.1
		N	76.8	94.2	62.9	83.8	42.0	69.5	71.5
GLAM [46]‡	$\mathcal{R}$GLDv2-clean	Y	76.4	89.4	69.3	85.2	48.9	74.2	73.9
		N	75.6	93.3	61.7	84.0	43.1	68.1	71.0
DOLG [47]	GLDv2-clean	Y	81.5	94.3	72.8	87.0	48.2	76.0	76.6
		N	75.7	93.1	62.7	82.1	42.0	64.4	70.0
DOLG [59]†	$\mathcal{R}$GLDv2-clean	Y	76.1	88.6	66.1	79.7	41.5	64.1	69.4
		N	74.6	91.9	61.1	82.0	37.1	65.0	68.6

Table A13. mAP comparison of the original GLDv2-clean training set with our revisited version $\mathcal{R}$GLDv2-clean separately for *overlapping classes (OC)* vs. *non-overlapping* for a number of SOTA methods. For GLDv2-clean, we evaluate pre-trained models. For $\mathcal{R}$GLDv2-clean we reproduce training with ResNet101 backbone, ArcFace loss and same sampling, settings and hyperparameters. †/‡: official/our code.

Method	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
ECNet [53]	88.2	91.5	66.8	78.3	42.0	55.4
NLNet [54]	89.4	91.8	66.5	77.6	39.1	53.7
Gather-Excite [12]	89.4	90.5	66.7	77.1	41.2	53.8
SENet [13]	**89.9**	**92.0**	**67.3**	**79.4**	**42.4**	**57.5**

Table A14. mAP comparison of different *backbone enhancement* (BE) options. Training on SfM-120k.

Method	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
ASPP [5]	**90.3**	92.2	**67.9**	78.2	41.6	55.8
SKNet [21]	89.3	**92.4**	67.4	78.4	42.3	55.5
SKNet†	89.9	92.0	67.3	**79.4**	**42.4**	**57.5**

Table A15. mAP comparison of different *selective context* (SC) options. Training on SfM-120k. SKNet†: our modification of SKNet [21].

SC	AL	Oxf5k	Par6k	Medium		Hard	
				$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
		87.1	90.6	63.9	77.3	36.7	53.9
✓		87.6	90.8	64.7	77.8	37.9	54.8
	✓	89.7	92.0	66.8	**79.4**	41.8	**57.5**
✓	✓	**89.9**	**92.0**	**67.3**	**79.4**	**42.4**	**57.5**

Table A16. mAP comparison of *learnable-fusion* (✓) *vs. sum*. Training on SfM-120k. SC: selective context; AL: attentional localization.

Backbone	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Attention-based pooling	89.8	**92.3**	67.2	**79.4**	41.8	56.5
Mask-based pooling (Ours)	**89.9**	92.0	**67.3**	**79.4**	**42.4**	**57.6**

Table A17. mAP comparison of pre-trained model with *attention-based* vs. *mask-based pooling*. Training on SfM-120k.

separately for overlapping *vs*. non-overlapping classes. That is, classes of the evaluation set that overlap or not with the original training set. As expected, mAP is much higher for overlapping than non-overlapping classes on GLDv2-clean. On $\mathcal{R}$GLDv2-clean, differences are smaller or even non-overlapping are higher.

D. More ablations

Fine-tuning We employ transfer learning from models pre-trained on ImageNet [42]. Following DIR [8] and DELF [28], we first perform classification-based training of the backbone only on the landmark training set and then fine-tuning the model on the same training set, training CiDeR while the backbone is kept frozen. Figure A10 visualizes this process, while Figure A9 shows the training and validation loss and accuracy, with and without the fine-tuning process. These plots confirm that fine-tuning results in lower loss and higher accuracy on both training and validation sets. This is corroborated by improved performance results (CiDeR +FT) in Table 6. Compared to the results without the fine-tuning, we obtain gains of 2.7% and 3.1% on Ox5k and Par6k Base, 8.9% and 5.1% on $\mathcal{R}$Oxf and $\mathcal{R}$Par Medium, and 16.5% and 11.4% on $\mathcal{R}$Oxf and $\mathcal{R}$Par Hard.

Backbone enhancement (BE) We apply four methods in a plug-and-play fashion [53, 54, 12, 13]. As shown in Table A14, SENet [13] performs best. We select it for backbone enhancement in the remaining experiments.

Selective context (SC) Here we compare ASPP [5], SKNet [21] and our modification, SKNet†. The modification is that instead of a simple *element-wise sum* to initially fuse multiple context information, we introduce a *learnable parameter* (5) to fuse feature maps based on importance. As shown in Table A15, our modification SKNet† performs best, confirming that this approach better embeds context information.

Sum (baseline) *vs*. learnable fusion We introduce learnable parameters (5) to fuse multiple feature maps for SC and AL. Table A16 compares this *learnable fusion* with simple *sum* for both SC and AL. We evaluate four different combinations, using learnable fusion and sum for SC and AL. The results indicate that learnable fusion improves performance wherever it is applied.

Attention-based *vs*. mask-based pooling Because of the binary masks (3), the pooling operation of our attentional localization (AL) can be called *mask-based pooling*. Here we derive a simpler baseline and connect it with attention in transformers. Given the feature tensor $\mathbf{F} \in \mathbb{R}^{w \times h \times d}$, we flatten the spatial dimensions to obtain the *keys* $K \in \mathbb{R}^{p \times d}$, where $p = w \times h$ is the number of patches. The weights of the 1×1 convolution f^ℓ can be represented by *query* $Q \in \mathbb{R}^{1 \times d}$, which plays the role of a learnable CLS token. Then, replacing the nonlinearity $\eta(\zeta(\cdot))$ by softmax, the spatial attention map (1) becomes

$$A = \text{softmax}(QK^\top) \in \mathbb{R}^{1 \times p}. \tag{A7}$$

Then, by omitting the masking operation and using the attention map A to weight the *values* $V = K \in \mathbb{R}^{p \times d}$, (4)

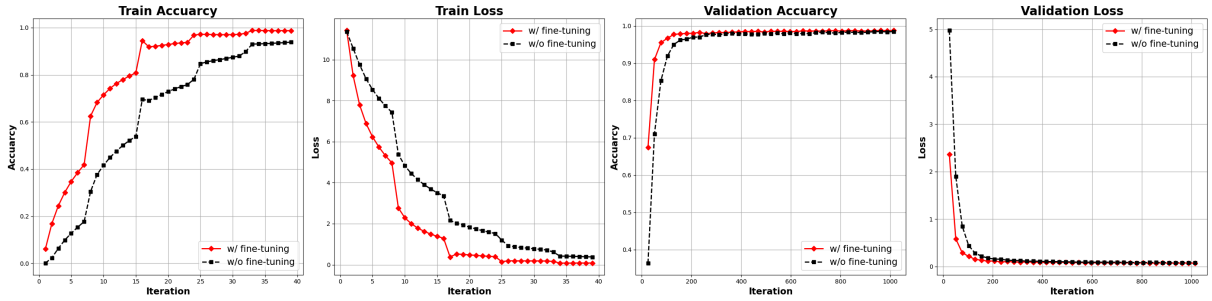


Figure A9. Comparison of the accuracy and loss for training and validation *with (red)* vs. *without (black) fine-tuning*.

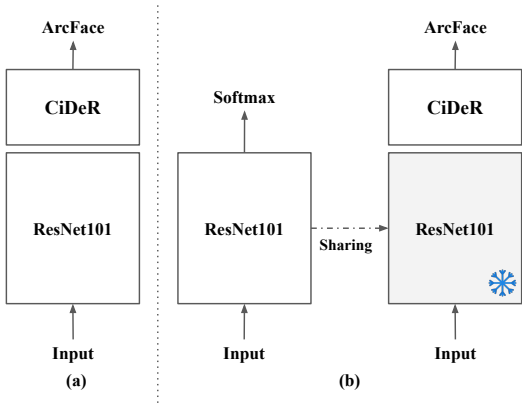


Figure A10. *Fine-tuning process*. (a) No fine-tuning. (b) Our fine-tuning. ❄: frozen.

simplifies to

$$\mathbf{F}^{\ell} = A^{\top} \odot V \in \mathbb{R}^{p \times d}, \tag{A8}$$

Finally, we apply spatial pooling f^{p}, like GAP or GeM. For example, in the case of GAP, the final pooled representation becomes

$$f^{p}(\mathbf{F}^{\ell}) = AV \in \mathbb{R}^{1 \times d}, \tag{A9}$$

which is the same as a simplified cross-attention operation between the features $\mathbf{F}$ and a learnable CLS token, without projections. By using GeM pooling, we refer to this baseline as *attention-based pooling*. Variants of this approach have been used, mostly for classification [19, 61, 37, 52, 33]. As shown in Table A17, our mask-based pooling is on par or performs better than the attention-based pooling baseline, especially on the hard protocol.

Feature dimension After applying spatial pooling like GeM, we apply an FC layer to generate the final features. The feature dimension d is a hyperparameter. Table A18 shows the performance for different dimensions d. Interestingly, a feature dimension of 2,048 works best, with larger dimensions not necessarily offering any more performance improvement.

Dim	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
4096	87.8	90.2	64.8	76.8	38.1	52.8
3097	89.8	90.5	**67.4**	76.9	**42.5**	53.0
2048	**89.9**	**92.0**	67.3	**79.4**	42.4	**57.5**
1024	88.9	91.2	65.7	76.7	40.1	52.0
512	85.3	89.2	61.9	74.4	36.5	48.9

Table A18. mAP comparison of different *feature dimensions* d in our model. Training on SfM-120k.

Query	Database	Oxf5k	Par6k	$\mathcal{R}$Medium		$\mathcal{R}$Hard	
				$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Single	Single	90.5	91.5	67.0	77.4	40.3	55.0
Multi	Single	**92.6**	**92.9**	**68.4**	79.2	41.2	56.5
Single	Multi	87.1	90.4	64.8	77.5	39.1	55.9
Multi	Multi	89.9	92.0	67.3	**79.4**	**42.4**	**57.5**

Table A19. mAP comparison using *multiresolution* representation (Multi) or not (Single) on query or database images. Training on SfM-120k.

Backbone	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Facebook	88.2	**92.7**	65.3	78.8	38.6	56.5
TorchVision	**89.9**	92.0	**67.3**	**79.4**	**42.4**	**57.5**

Table A20. mAP comparison of pre-trained model from *TorchVision* vs. *Facebook*. Training on SfM-120k.

Multi-resolution At inference, we use a multi-resolution representation at image scales (0.4, 0.5, 0.7, 1.0, 1.4) for both the query and the database images. Features are extracted at each scale and then averaged to form the final representation. Table A19 provides a comparative analysis, with and without the multi-resolution representation for query and database images. We find that applying multi-resolution to both query and database images works best for $\mathcal{R}$Oxford and $\mathcal{R}$Paris [34].

ImageNet pre-trained models Different research teams have released models pre-trained on ImageNet [42] for major image classification tasks. It is common to use a pre-trained ResNet101 model from TorchVision [24]. Recent works [59,

WARM-UP	OXF5K	PAR6K	MEDIUM		HARD	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
	89.9	**92.0**	66.7	78.9	41.5	56.7
✓	**89.9**	**92.0**	**67.3**	**79.4**	**42.4**	**57.5**

Table A21. mAP effect of *warm-up* in our model training. Training on SfM-120k.

WHITENING	OXF5K	PAR6K	MEDIUM		HARD	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
	85.8	90.7	60.5	77.0	31.7	54.2
✓	**89.9**	**92.0**	**67.3**	**79.4**	**42.4**	**57.5**

Table A22. mAP effect of *whitening* in our model. Training on SfM-120k.

METHOD	OXF5K	PAR6K	$\mathcal{R}$MEDIUM		$\mathcal{R}$HARD	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Fixed-size (224×224)	69.0	86.0	42.2	69.5	15.8	45.0
Group-size (Ours)	**89.9**	**92.0**	**67.3**	**79.4**	**42.4**	**57.5**

Table A23. mAP comparison between *fixed-size* (224×224) *vs. group-size sampling*. Training on SfM-120k.

20] have also used pre-trained models released by Facebook[3]. As shown in Table A20, we find that the TorchVision model works best.

Warm-Up To enhance model performance, we employ a warm-up phase [11] during training, consisting of three epochs. Table A21 shows that the warm-up phase improves the performance.

Whitening We utilize the supervised whitening method pioneered by Radenović *et al*. [36], which is common in related work to improve retrieval performance. Table A22 shows the performance gain obtained by the application of whitening.

Fixed-size *vs*. group-size sampling Several previous studies suggest organizing training batches based on image size for efficient learning. Methods such as DIR [8], DELF [28], MobileViT [25], and Yokoo *et al*. [60] opt for variable image sizes rather than adhering to a single, fixed dimension. Our approach employs group-size sampling [60, 46, 47], where we construct image batches with similar aspect ratios. Table A23 compares the results of fixed-size (224×224) and group-size sampling. We find that using dynamic input sizes to preserve the aspect ratio significantly improves performance.

[3] https://github.com/facebookresearch/pycls