

# On the Exploration of Incremental Learning for Fine-grained Image Retrieval

Wei Chen<sup>1</sup>

w.chen@liacs.leidenuniv.nl

Yu Liu<sup>2</sup>

yu.liu@esat.kuleuven.be

Weiping Wang<sup>3</sup>

wangwp@nudt.edu.cn

Tinne Tuytelaars<sup>2</sup>

Tinne.Tuytelaars@esat.kuleuven.be

Erwin M. Bakker<sup>1</sup>

e.m.bakker@liacs.leidenuniv.nl

Michael Lew<sup>1</sup>

m.s.k.lew@liacs.leidenuniv.nl

<sup>1</sup> MediaLab, LIACS,

Leiden University,

Leiden, The Netherlands

<sup>2</sup> ESAT-PSI,

KU Leuven,

Belgium

<sup>3</sup> College of Systems Engineering,

NUDT,

Changsha, China

In this supplementary material, we provide additional quantitative and qualitative results, which are not shown in the main paper. We tested on another dataset: Stanford-Dogs [10]. Note that recorded results are under the same configurations with the CUB-Birds dataset.

## 1 One-step Incremental Learning for FGIR

### (1) Recall@K Evaluation on the Stanford-Dogs Dataset

The process includes two stages. First, we use the cross-entropy and triplet loss to train the network  $\mathcal{A}$  on the original classes (1-60), denoted as  $\mathcal{A}(1-60)$ . Second, only images of new classes are added at once to train network  $\mathcal{B}$ , denoted as  $\mathcal{B}(61-120)$ . We observe similar trends as the results we shown in main paper, when our method achieves good performance on the original classes and new classes with Recall@1= 76.67% and Recall@1=81.88%, respectively. Compared to the initial model on the original classes, our method has dropped Recall@1 performance by 4.00% (80.67%→76.67%).

### (2) Precision-Recall Curves and mAP Results

We report the precision-recall curves and mAP results in Figure 1. We can observe these curves share with the similar trends with those from the CUB-Birds dataset. Overall, our method can effectively address the catastrophic forgetting issue on the original classes while achieve ideal performance on the new classes.

### (3) t-SNE Visualization for Feature Distribution

We visualize the feature distributions with and without MMD loss in Figure 2, which demonstrate the MMD loss reduces the distance between distributions and effectiveness for mitigating the forgetting issue.

Configurations Recall@K(%)	Original classes			New classes		
	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}(1-60)$ (initial model)	80.67	87.27	92.20	-	-	-
+ $\mathcal{B}(61-120)$ w feature extraction	-	-	-	75.64	83.91	90.48
+ $\mathcal{B}(61-120)$ w fine-tuning	61.43	72.80	81.70	78.93	86.99	91.55
+ $\mathcal{B}(61-120)$ w LwF ( $L_{dist}$ )	61.77	72.72	81.70	78.52	86.38	91.12
+ $\mathcal{B}(61-120)$ w EWC	62.24	73.30	82.82	78.90	86.59	91.19
+ $\mathcal{B}(61-120)$ w ALASSO	62.61	74.49	82.98	78.14	85.98	91.02
+ $\mathcal{B}(61-120)$ w L2 loss	72.07	81.44	87.47	<b>82.21</b>	88.75	92.52
+ $\mathcal{B}(61-120)$ w Our method	<b>76.67</b>	<b>85.10</b>	<b>91.14</b>	81.88	<b>88.98</b>	<b>93.36</b>
$\mathcal{A}(1-120)$ (reference model)	79.29	86.86	91.61	82.57	88.75	93.13

Table 1: Recall@K (%) of incremental FGIR on the Stanford-Dogs dataset when new classes are added at once. The best performance are in bold.

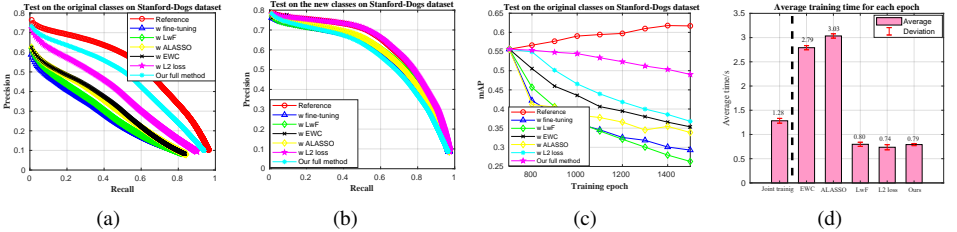


Figure 1: Figure (a)-(b) denote the precision-recall curves tested on the original classes and new classes on the Stanford-Dogs dataset. The larger the area under each curve, the better performance of the method. Figure (c) depicts the mAP results for different methods as the training proceeds. We only show the results tested on the original classes. Being closer to the reference curve (red one) indicates less performance degradation, *i.e.*, the method can maintain its previous performance on the original classes on the Stanford-Dogs dataset.

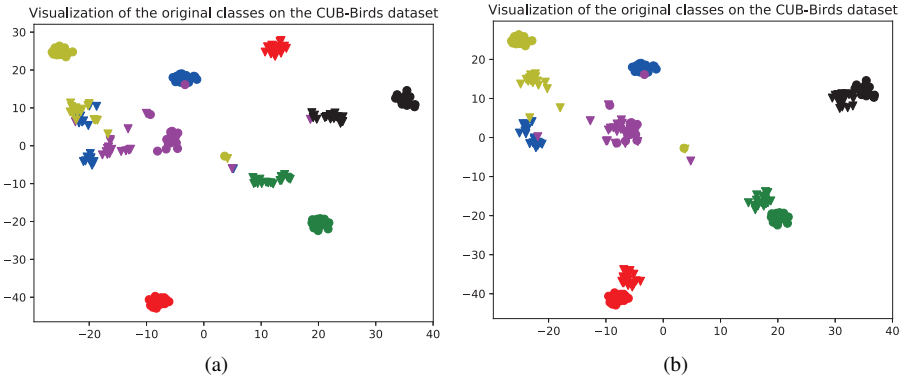


Figure 2: t-SNE visualization for feature distribution of 6 categories. The circle shape indicates the features from reference model, which has the same distribution in two cases. The triangle shape denotes the feature from models trained with/without MMD loss. (a): model trained without MMD loss; (b): model trained with MMD loss.

## 2 Influence of Added Multiple Classes

In previous experiments, we add multiple classes (*i.e.* 100 new classes for the CUB-Birds dataset) for one-step incremental training at once. Herein, we further explore the influence of the new classes number for the Stanford-Dogs dataset where we choose 60 new classes and 5 classes for incremental learning.

The results are reported in Table 2. We observe these two datasets share with similar trends that larger new coming classes lead to heavier forgetting issue. For the Stanford-Dogs dataset, when only 5 new classes are added, the Recall@1 drops from 80.67% to 79.75%, compared to the one drops from 80.67% to 76.67% when 60 new classes are added.

Configurations	Original classes			New classes <sup>†</sup>		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}(1-60)$ (initial model)	80.67	87.27	92.20	-	-	-
+ $\mathcal{B}(61-65)$ w Our full method	79.75	87.23	91.92	97.45	98.55	99.27
+ $\mathcal{B}(61-120)$ w Our full method	76.67	85.10	91.14	81.88	88.98	93.36
$\mathcal{A}(1-65)$ (reference model)	79.62	86.15	90.91	96.73	97.82	98.55
$\mathcal{A}(1-120)$ (reference model)	79.29	86.86	91.61	82.57	88.75	93.13

Table 2: Recall@K (%) on the Stanford-Dogs dataset when 5 or 60 new classes are added at once. Correspondingly, <sup>†</sup> indicates the results are tested on different new classes.

### 3 Multi-step Incremental Learning for FGIR

We report the results on the Stanford-Dogs dataset in Table 3 when new classes are added sequentially. We observe similar trends as those for the CUB-Birds dataset. Compared to the other two methods, the proposed method has ideal retrieval performance on the newly added classes and the original classes.

Configurations		Original (1-60)			Added new (61-75)			Added new (76-90)			Added new (91-105)			Added new (106-120)		
Recall@K(%)		K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}(1-60)$ (initial model)		80.67	87.27	92.20	-	-	-	-	-	-	-	-	-	-	-	-
LwF algorithm [10]	+ $\mathcal{B}(61-75)$	50.87	62.76	73.40	88.35	92.48	94.36	-	-	-	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)$	42.06	53.62	65.10	71.18	82.46	88.10	77.33	87.19	92.44	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)$	37.58	50.48	63.00	60.65	73.68	82.33	70.10	81.38	87.40	80.62	87.17	92.48	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)(106-120)$	38.46	50.63	62.59	59.90	72.68	81.20	63.86	77.22	85.54	68.41	77.70	85.66	81.34	88.69	92.74
EWC algorithm [11]	+ $\mathcal{B}(61-75)$	55.84	67.64	77.57	89.10	92.61	94.36	-	-	-	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)$	45.32	58.29	68.85	79.82	85.21	90.35	81.38	88.72	93.32	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)$	37.60	49.71	61.88	67.04	79.04	85.71	67.47	79.52	88.39	81.33	86.99	91.15	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)(106-120)$	34.08	45.60	58.40	63.53	75.19	83.71	63.42	77.66	86.31	70.00	79.12	85.84	81.99	87.96	92.10
L2 loss algorithm [12]	+ $\mathcal{B}(61-75)$	65.30	75.83	83.51	90.85	94.74	95.61	-	-	-	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)$	55.97	67.36	77.04	84.46	90.73	92.73	80.94	89.38	93.54	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)$	50.38	62.87	73.64	72.68	82.21	88.85	75.68	84.67	91.57	83.72	90.00	93.81	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)(106-120)$	46.01	58.74	69.64	67.79	78.07	85.71	72.51	84.45	90.03	74.87	83.98	89.82	86.21	91.36	94.39
Our method	+ $\mathcal{B}(61-75)$	76.07	84.88	90.11	91.85	95.36	96.87	-	-	-	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)$	70.67	80.48	87.87	89.10	93.11	95.99	84.23	89.92	93.43	-	-	-	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)$	67.75	79.17	86.45	86.09	91.98	95.49	81.60	90.25	93.76	84.25	89.03	93.45	-	-	-
	+ $\mathcal{B}(61-75)(76-90)(91-105)(106-120)$	65.47	76.52	85.08	83.21	89.35	93.73	79.19	87.84	93.32	82.83	89.20	94.42	87.13	92.10	94.39
$\mathcal{A}(1-120)$ (reference model)		79.29	86.86	91.61	92.61	94.99	96.37	82.48	90.80	93.76	83.72	91.33	95.58	86.12	93.11	95.96

Table 3: Recall@K (%) results on the Stanford-Dogs dataset when new classes are added sequentially. “Added new (61-75)” indicates we use first 15 classes (61-75) as the first incremental part to train the network.

## References

- [1] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop*, volume 2, 2011.
- [2] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *PNAS*, 114 (13):3521–3526, 2017.

- [3] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017.
- [4] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCV Workshops*, 2019.