Supplementary material for "Boosting vision transformers for image retrieval"

Chull Hwan Song[1] Jooyoung Yoon[1] Shunghyun Choi[1] Yannis Avrithis[2,3]
[1]Dealicious Inc. [2]Institute of Advanced Research on Artificial Intelligence (IARAI) [3]Athena RC

A. More experiments

A.1. Setup

Network size and pretraining The ViT-B transformer encoder and the R50+ViT-B hybrid have 86M and 98M parameters respectively and are pretrained on ImageNet-21k. The additional components of DToP have only 0.3M parameters. The common competitors Resnet50/101 have 25M and 44M parameters respectively and are pretrained on ImageNet-1k. Despite its lower dimensionality (1,536 *vs*. 2,048 dimensions for Resnet101), our DToP-R50+ViT-B is of course in a priviledged position over Resnet101 in terms of both network size and pretraining data. Still, it is important that a transformer reaches SOTA on image retrieval for the first time. Our objective is not to introduce a new architecture.

Training settings We apply random cropping, illumination and scaling augmentation. We use a batch size of 64 and the ArcFace loss with margin 0.15. We optimize by stochastic gradient descent with momentum 0.9, initial learning rate 10^{-3} and cosine learning rate decay with the decay factor 10^{-4}. We apply warm-up of 0, 3, 5 and 5 epochs on NC-clean, SfM-120k, GLDv1-noisy and GLDv2-clean, respectively. We implement our method on PyTorch and we train our models on 8 TITAN RTX 3090Ti GPUs.

A.2. Benchmarking of vision transformer models

Vision transformer studies have exploded in a short period of time, but very few concern image retrieval. We perform for the first time an extensive empirical study to benchmark a large number of vision transformer models on image retrieval and choose the best performing one as our default backbone.

Candidate models We fine-tune on image retrieval training sets, so we only consider models that are pre-trained on ImageNet-1k or ImageNet-21k [13]. In particular, we consider the models shown in Table A4.

As global image representation, all models use a `[CLS]` (classification) token embedding, while PiT [26] and DeiT [28] also provide a distillation token embedding. As a local image representation, patch token embeddings can be used for all models, while for the hybrid R50+ViT-B, features of the CNN stem can also be used. Certain pre-trained models, in particular Swin [40], ConViT [12] and TNT [22], cannot handle multi-scale input. These models are not compatible with the group-size sampling approach that we adopt for training [73]. This constraint is not due to the architecture itself but to the way the code is written, and although it would be certainly possible to fix, this would require some effort. We therefore exclude them from our benchmark.

MODEL	CLS	DIST	PATCH	CNN	MS
Swin [40]	✓		✓		
ConViT [12]	✓		✓		
TNT [22]	✓		✓		
ViL [76]	✓		✓		✓
CvT [67]	✓		✓		✓
LocalViT [35]	✓		✓		✓
Patch [18]	✓		✓		✓
T2T [75]	✓		✓		✓
DeepViT [77]	✓		✓		✓
LV-ViT [30]	✓		✓		✓
PiT [26]	✓	✓	✓		✓
DeiT (DeiT-B) [28]	✓	✓	✓		✓
ViT (ViT-B) [32]	✓		✓		✓
ViT (R50+ViT-B)	✓		✓	✓	✓

Table A4: Feature types that can be extracted from different pre-trained vision transformer models considered in our study. CLS: classification token; Dist: distillation token; Patch: patch tokens. CNN: convolutional stem (hybrid model). MS: can handle multi-scale input at training (required in our experiments).

Setup We take all models as pre-trained on either ImageNet-1k or ImageNet-21k and fine-tune them on SfM-120k [52]. We use only the global branch (7) on multi-layer `[CLS]` features (5) or the local branch (10) on multi-layer patch features (6). In the former case, we also evaluate multi-layer *distillation* features, replacing `[CLS]` by the distillation token where it exists, *i.e*., PiT [26] and DeiT [28]. In the latter case, we do not use the enhanced locality module (ELM), that is, we set $Y' = Y$ in (9). At the output, instead of (12), we only use an FC layer with output feature dimension $N = 768$. We use group-size sampling [73] with our dynamic position embedding (DPE) (2). We use default training settings, except without warm-up.

At inference, we evaluate each model with exactly the

same type of features as at training ([CLS], distillation or patch), applying supervised whitening [52] on multi-scale features [52] and measuring mAP on the evaluation sets.

Results Table A5 shows the results of the benchmark. In the majority of cases, we can see that [CLS] outperforms patch features: CvT-21 [67], LocalViT-S [35], LV-ViT-M [30], DeiT-B [28], ViT-B [32], R50+ViT-B [32]. However, in many cases, the opposite holds: Patch-ViT-B [18], T2T-ViT-24 [75], DeepViT-B [77]. In few cases, the performance is similar or inconsistent: ViL-B [76], PiT-B [26]. As for the distillation token, it works consistently better than [CLS] for DeiT-B [28], but for PiT-B [26] there is no clear winner. Overall, we observe that when multi-layer features are used for image retrieval, [CLS] does not work necessarily better than global average pooling. ImageNet-21k is also not necessarily better than the smaller ImageNet-1k as a training set; for example, Patch-ViT-B [18] using [CLS] performs worse overall using this training set.

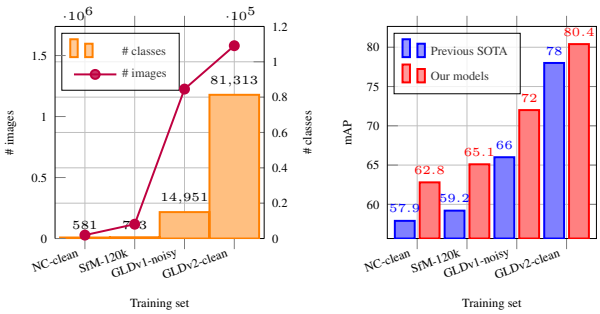
Using the [CLS] token, the hybrid model R50+ViT-B [32] is a clear winner overall, also better than the same model using patch features. The second and third best are CvT-21 [67] using [CLS] and DeiT-B [28] using distillation token, respectively. The hybrid model has more parameters (98M) than the plain ViT-B [32]/DeiT-B [28] (86M) and a lot more than CvT-21 [67] (31M) and the majority of models. It is also pretrained on the larger ImageNet-21k training set. In this sense, this is not a fair comparison. There are many other factors that are not shown here, like distillation from a stronger model by DeiT-B [28] and PiT-B [26].

We choose R50+ViT-B [32] as the default backbone in our experiments for two reasons: (a) Our objective is to explore how much vision transformers can improve on image retrieval; and (b) its improvement over other models on the Hard protocol is more pronounced, implying it is a much stronger model. We suspect that its improvement is also significant in the presence of the challenging $\mathcal{R}$1M distractors, although we cannot benchmark all models for this. What we can suggest as a more lightweight alternative is CvT-21 [67].

A.3. More results

Summary of progress per dataset Table A6(a) and Figure A5(a) show statistics of open datasets that have been used as training sets for landmark image retrieval; in particular, number of classes and number of images per dataset. There is a variety of dataset sizes, both in terms of classes and images. In our experiments, we focus on the most commonly used datasets in the literature, that is, *neural code* (NC) clean [19], *structure-from-motion* 120k (SfM-120k) [52], *Google landmarks v1* (GLDv1) noisy [46] and *Google landmarks v2* (GLDv2) clean [65].

Table A6(b) and Figure A5(b) show the progress over the SOTA that we bring per dataset, based on global features. We compare separately per training set, in terms of mAP averaged over four columns of Table 1. We observe that as the number of training images increases, the performance also increases. Our models bring clear improvement on all training sets, with the improvement being more pronounced on small and noisy training sets.



(a) Training set statistcs (b) Our *vs*. previous models

Figure A5: (a) Open landmark training set statistics, from Table A6(a). (b) Average mAP comparison of our *vs*. previous SOTA models based on global features. Average taken over $\mathcal{R}$Oxf and $\mathcal{R}$Par test sets in $\mathcal{R}$Medium and $\mathcal{R}$Hard protocols (four columns) in Table 1. Previous SOTA models are as shown in Table A6(b).

A.4. More ablation experiments

Like subsection 4.4, all experiments here are conducted on SfM-120k by default. We investigate the effect of different factors on the performance of our full model DToP-R50+ViT-B, including the number of layers k, DPE against baselines, ELM components, the feature dimension N, multi-scale features at inference, mini-batch sampling as well as fusion alternatives for (9).

Number of layers Table A8 shows the effect of the number of layers k in the multi-layer features (5),(6). The optimal number of layers is $k = 6$. The effect of k is significant, especially in the Hard protocol, yielding an improvement of 6.1% mAP over the baseline $k = 1$ on $\mathcal{R}$Oxf. What is less understood is that $k = 12$ works as well.

Dynamic position embedding In Table A9 we compare our DPE with baselines and assess the effect of interpolation type we use in DPE. Bi-linear interpolation is best, outperforming the baseline by nearly 20% on $\mathcal{R}$Par Hard.

ELM components In Table A10 we study the effect of different components of the enhanced locality module (ELM). It is clear that all components contribute to the performance of ELM in the absence of the CNN stem. In the hybrid architecture, they are effective on Oxf5k and $\mathcal{R}$Oxf but not on Par6k and $\mathcal{R}$Par. This result is in agreement with Table 3, where ELM is most effective in the absence of other forms of inductive bias.

Feature dimension The global representation obtained by our DToP model is given by (12) and is a vector of dimension

Model	Pre-Train	Params (M)	Global (CLS/Dist)							Local (Patch)					
			Token	Oxf5k	Par6k	Medium		Hard		Oxf5k	Par6k	Medium		Hard	
						$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
T2T-ViT-24 [75]	ImageNet-1k	64	CLS	35.3	49.3	14.2	35.5	2.0	10.6	65.0	72.0	42.0	52.1	18.6	22.8
DeepViT-B [77]	ImageNet-1k	48	CLS	43.9	56.9	21.1	41.7	4.2	13.4	50.4	62.1	27.7	46.2	7.1	18.2
LocalViT-S [35]	ImageNet-1k	22	CLS	65.0	71.3	38.4	53.0	14.9	22.1	62.1	71.4	36.5	50.9	11.2	19.1
LV-ViT-M [30]	ImageNet-1k	39	CLS	72.4	79.2	45.4	65.2	19.6	28.9	42.1	57.1	25.6	45.8	11.4	20.9
Patch-ViT-B [18]	ImageNet-21k	86	CLS	28.8	38.7	13.5	28.5	1.7	8.0	59.5	68.9	34.5	50.6	13.1	18.6
ViL-B [76]	ImageNet-21k	55	CLS	42.6	56.0	20.7	39.5	3.1	12.8	43.6	51.1	20.1	36.4	3.1	10.8
CvT-21 [67]	ImageNet-21k	31	CLS	84.0	87.8	61.3	78.8	30.8	56.7	77.6	81.6	54.6	**76.4**	25.7	**52.9**
PiT-B [26]	ImageNet-1k	73	CLS	69.0	77.0	43.6	59.1	19.8	28.7	70.9	81.2	42.4	64.8	18.9	35.5
			Dist	66.1	78.5	41.5	60.6	15.5	30.1						
DeiT-B [28]	ImageNet-1k	86	CLS	82.3	84.1	59.6	67.6	29.1	46.5	**80.7**	78.5	**54.7**	64.5	22.7	37.2
			Dist	84.2	85.5	62.4	71.2	32.6	49.7						
ViT-B [32]	ImageNet-21k	86	CLS	76.2	83.4	49.3	70.9	19.6	46.4	60.4	81.0	38.6	69.2	12.0	46.9
R50+ViT-B [32]	ImageNet-21k	98	CLS	**84.3**	**87.9**	**62.6**	**79.6**	**37.9**	**64.8**	74.6	**82.8**	50.9	69.4	**26.9**	46.5

Table A5: mAP comparison of different pre-trained vision transformer models, using multilayer `[CLS]` or distillation (Dist) token features from our global branch (7), or multi-layer patch features from the local branch (10), without ELM (11). All options give rise to a global representation of $N = 768$ dimensions; patch features undergo global average pooling. Fine-tuning on SfM-120k [52] using default settings.

Train Set	(a) Statistics		(b) Improvement			
	#Classes	#Images	Previous SOTA		Ours	Gain
NC-noisy [2]	672	213,678				
NC-clean [19]	581	27,965	RMAC [19, 50]	57.9	62.8	+4.9
SfM-120k [52]	713	117,369	GeM [52, 50]	59.2	65.1	+5.9
GLDv1-noisy [46]	14,951	1,225,029	SOLAR [43]	66.0	72.0	+6.0
GLDv2-noisy [65]	203,094	4,132,914				
GLDv2-clean [65]	81,313	1,580,470	DOLG [72]	78.0	80.4	+2.4

Table A6: (a) Open landmark training set statistics. Bold: datasets used in our experiments. (b) Average mAP comparison of our *vs*. previous SOTA models based on global features. Average taken over $\mathcal{R}$Oxf and $\mathcal{R}$Par evaluation sets in $\mathcal{R}$Medium and $\mathcal{R}$Hard protocols (four columns) in Table 1.

Dim N	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
128	80.7	89.1	56.7	78.5	29.6	59.3
256	87.4	90.1	63.1	80.2	35.8	62.2
512	88.0	91.1	64.9	82.1	38.6	64.7
768	88.8	91.8	67.5	82.1	41.7	64.8
1,024	86.3	92.7	64.4	**83.1**	38.4	**66.5**
1,536	**89.7**	92.7	**68.5**	**83.1**	**43.0**	65.8
2,048	89.3	**93.0**	65.8	82.9	39.0	65.8

Table A7: mAP comparison of different dimension N of output features (12) of our full model. Training on SfM-120k. Using supervised whitening [52].

N. Table A7 shows the effect of the choice of this dimension. Clearly, the best performance is obtained by $N = 1,536$, by a larger margin on $\mathcal{R}$Oxford (medium or hard). A larger dimension does not necessarily mean better performance.

Multi-scale Following previous work [19, 52], we use a multi-scale image representation with 3 scales at inference by computing the output features (12) for each scale of the input image and averaging the features over scales. Table A11 shows the effect of using multi-scale *vs*. single-scale representation on queries or database. Clearly, using a multi-scale representation on the database works best. As for the queries, the results are not consistent across datasets, but the gain brought by multi-scale queries on $\mathcal{R}$Paris is more than the loss on $\mathcal{R}$Oxford. We thus choose a multi-scale representation on both queries and database.

Layers k	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
1	87.2	92.4	64.9	81.3	37.6	62.7
3	88.0	91.1	64.9	82.1	38.6	64.7
6	**89.7**	92.7	**68.5**	**83.1**	**43.0**	**65.8**
9	87.1	**93.1**	66.8	82.4	41.4	64.2
12	89.0	92.4	68.1	83.0	43.2	65.5

Table A8: mAP comparison of using different number of layers k in the multi-layer features (5),(6). Training on SfM-120k.

PE Type	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
no PE	82.8	85.7	59.7	73.9	32.5	47.4
CPE [11]	85.9	88.8	62.6	77.9	37.1	58.2
DPE (bi-cubic)	87.6	91.0	65.2	82.2	38.3	64.6
DPE (bi-linear)	**89.7**	**92.7**	**68.5**	**83.1**	**43.0**	**65.8**

Table A9: mAP comparison of our (bi-linear and bi-cubic) *dynamic position embedding* (DPE) (2) with no position embedding and with conditional position embedding (CPE) [11]. Training on SfM-120k.

Mini-batch sampling As discussed in subsection 3.2, we use group-size sampling [73] to account for different sizes and aspect ratios of input images, while maintaining the same size for all images in a mini-batch. This strategy results

CNN Stem	IRB	ASPP	WB	Oxf5k	Par6k	Medium		Hard	
						$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
		✓	✓	78.6	87.8	55.2	77.4	27.1	55.1
	✓		✓	75.8	87.8	52.9	77.5	28.7	53.7
	✓	✓		80.1	88.2	59.8	77.7	28.8	54.2
	✓	✓	✓	81.5	89.8	61.4	79.7	32.5	57.4
✓		✓	✓	84.9	92.7	64.8	**83.4**	42.4	65.7
✓	✓		✓	85.3	**93.0**	65.7	83.0	42.8	65.4
✓	✓	✓		87.0	92.0	66.5	82.7	42.8	65.1
✓	✓	✓	✓	**89.7**	92.7	**68.5**	83.1	**43.0**	**65.8**

Table A10: mAP comparison of variants of enhanced locality module (ELM) with/without different components. IRB: inverted residual block [55]; ASPP: à trous spatial pyramid pooling [7]; WB: WaveBlock [63]. Training on SfM-120k.

Query	Database	Oxf5k	Par6k	Medium		Hard	
				$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Single	Single	89.4	92.2	68.2	82.2	43.0	64.0
Multi	Single	**90.1**	92.5	68.6	82.8	43.1	65.1
Single	Multi	89.4	92.4	**68.8**	82.6	**43.8**	64.9
Multi	Multi	89.7	**92.7**	68.5	**83.1**	43.0	**65.8**

Table A11: mAP comparison of multi-scale *vs*. single-scale representation on queries or database. Training on SfM-120k.

Sampling	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Fixed-size	83.2	90.6	60.5	79.5	35.7	59.8
Group-size [73]	**89.7**	**92.7**	**68.5**	**83.1**	**43.0**	**65.8**

Table A12: mAP comparison of fixed-size (384×384) *vs*. group-size sampling [73] of mini-batches at training. Training on SfM-120k.

Method	Oxf5k	Par6k	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
No fusion (w/o ELM)	89.8	91.2	67.6	81.1	40.7	62.5
No fusion (w/ ELM)	85.9	91.9	64.6	82.6	40.6	65.4
Sum	89.5	91.7	68.4	82.1	43.4	64.2
Hadamard product	**89.8**	92.0	68.2	**83.1**	43.5	66.0
Concatenation	88.8	92.5	67.5	82.7	**43.9**	64.9
Fast normalized [59]	**89.8**	92.1	**68.7**	82.4	**43.9**	65.0
Orthogonal [72]	89.7	**92.7**	68.5	**83.1**	43.0	**65.8**

Table A13: mAP comparison of different feature fusion methods for the input and output of ELM (9). Training on SfM-120k.

in a dynamic image size per mini-batch and we use our dynamic position embedding (2) in this case. Table A12 shows the performance of this strategy compared with fixed-size (384×384) images for all mini-batches. It is clear that group-size sampling improves performance by a large margin, up to 8% on $\mathcal{R}$Oxford and up to 6% on $\mathcal{R}$Paris.

Feature fusion for ELM In the local branch, the patch features $Y \in \mathbb{R}^{w \times h \times D}$ (8) are fused in (9) with the output of the *enhanced locality module* (ELM), say, $U = \text{ELM}(Y) \in \mathbb{R}^{w \times h \times D}$. This happens because the input still has the valuable spatial information. Denoting function FUSE by h for brevity, Eq. (9) is written as

$$Y' = h(Y, U). \tag{A13}$$

Here we consider a number of alternatives for h:

$$\text{No fusion (w/o ELM)} : \quad h(Y, U) = Y \tag{A14}$$

$$\text{No fusion (w/ ELM)} : \quad h(Y, U) = U \tag{A15}$$

$$\text{Sum} : \quad h(Y, U) = Y + U \tag{A16}$$

$$\text{Hadamard product} : \quad h(Y, U) = Y \odot U \tag{A17}$$

$$\text{Concatenation} : \quad h(Y, U) = [Y; U] \tag{A18}$$

$$\text{Fast normalized [59]} : \quad h(Y, U) = \frac{w_1 Y + w_2 U}{w_1 + w_2 + \epsilon} \tag{A19}$$

$$\text{Orthogonal [72]} : \quad h(\mathbf{y}_i, \mathbf{u}_i) = [\mathbf{y}_i - \text{proj}_{\mathbf{u}_i}(\mathbf{y}_i); \mathbf{u}_i] \tag{A20}$$

Fast normalized fusion is a fusion strategy investigated as part of BiFPN [59], where $h(Y, U)$ is a linear combination of Y, U with $w_1 = \text{relu}(v_1)$, $w_2 = \text{relu}(v_2)$ and v_1, v_2 are learnable parameters. It has similar effect to normalizing v_1, v_2 by softmax but is more efficient.

Orthogonal fusion is similar to DOLG [72] but differs in that we fuse two 3D tensors while DOLG fuses a 3D tensor with a vector. By representing Y by a folded sequence of token embeddings $\mathbf{y}_1, \ldots, \mathbf{y}_M \in \mathbb{R}^D$ and similarly U by $\mathbf{u}_1, \ldots, \mathbf{u}_M \in \mathbb{R}^D$, we define $h(\mathbf{y}_i, \mathbf{u}_i)$ per token as the Gram-Schmidt orthogonalization of vectors $\mathbf{y}_i, \mathbf{u}_i$, where

$$\text{proj}_{\mathbf{u}}(\mathbf{y}) = \frac{\langle \mathbf{y}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u} \tag{A21}$$

is the orthogonal projection of $\mathbf{y}$ onto the line spanned by vector $\mathbf{u}$.

Table A13 shows a comparison of two non-fusion and the five fusion methods. We first observe that the output U of ELM alone can be worse than the input Y, while the five fusion methods mostly outperform the non-fusion variants. Hence, Y and U contain complementary information and fusion is beneficial. Of the fusion methods, the Hadamard product, fast normalized and orthogonal are the most effective, but there is no clear winner and differences are small. We choose orthogonal fusion as default.

A.5. More visualizations

Figure A6 provides t-SNE visualization of embeddings of $\mathcal{R}$Paris [50] by different models trained on SfM-120k [52]. It is clear that the class distribution of positive images under hard protocol are more overlapping than easy. Medium, being the union of easy and hard, is more populated but classes are better separated than in hard. Vision transformers clearly separate classes better than Resnet101, especially under hard protocol. The difference between our DToP-R50+ViT-B and the baseline hybrid model R50+ViT-B is small.

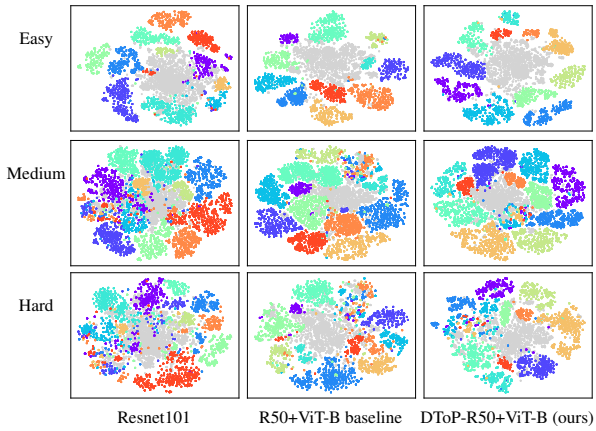


Figure A6: Visualization of *revisited Paris* ($\mathcal{R}$Paris or $\mathcal{R}$Par) evaluation set under *easy*, *medium* and *hard* protocols [50] (in rows) using t-SNE on output embeddings (12) obtained by different models (in columns) fine-tuned on SfM-120k [52]. For each protocol, positive images are colored by query group label and negative are gray. The set of medium positives is the union of easy and hard positives.