ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094987

MABNET: MASTER ASSISTANT BUDDY NETWORK WITH HYBRID LEARNING FOR IMAGE RETRIEVAL

Rohit Agarwal [†‡], *Gyanendra Das*[†§], *Saksham Aggarwal*[†§], *Alexander Horsch*[‡], *Dilip K. Prasad*[‡]

[‡]Bio-AI Lab, UiT The Arctic University of Norway, Tromsø, Norway
[§]Indian Institute of Technology (Indian School of Mines), Dhanbad, India

ABSTRACT

Image retrieval has garnered a growing interest in recent times. The current approaches are either supervised or self-supervised. These methods do not exploit the benefits of hybrid learning using both supervision and self-supervision. We present a novel Master Assistant Buddy Network (MABNet) for image retrieval which incorporates both the learning mechanisms. MABNet consists of master and assistant block, both learning independently through supervision and collectively via self-supervision. The master guides the assistant by providing its knowledge base as a reference for self-supervision and the assistant reports its knowledge back to the master by weight transfer. We perform extensive experiments on the public datasets with and without post-processing.

***Index Terms*—** Image Retrieval, Supervision, Self-Supervision, MABNet, ViT

1. INTRODUCTION

Image retrieval refers to the task of returning relevant instances from a database given an unlabeled query image. This task can be targeted in both supervised and self-supervised manners. Supervised methods forms better decision boundaries owing to the ground truth and are commonly based on Convolutional Neural Network (CNN) like ResNets [1–12] since they can extract image-level descriptors.

Recently, self-supervised learning has emerged to address the image retrieval task [13, 14]. DINO [13] exploits the attention mechanism of Vision Transformer (ViT) for self-supervision such that its knowledge model contains explicit information about the semantic segmentation of an image.

Supervision and self-supervision has their own advantages and the current literature has not explored the use of both in image retrieval task. Self-supervision will complement the supervised approach with explicit semantic segmentation improving the decision boundaries. We propose Master Assistant Buddy Network (MABNet), a novel model that employs both supervised and self-supervised learning mechanisms, thereby incorporating the advantages of both learning paradigm. MABNet is a two-block buddy network (see Fig. 1), where one is master and the other is assistant. Both learn individually via supervised learning. In addition, they compare their latent features for self-supervision using a distance metric where the assistant uses the master's latent features as a reference for self-supervision. The weights of the master block are later updated by the assistant's knowledge model via weight transfer. In this sense, the assistant network shares the learning load, performs self-supervision, and assists the master with a comprehensive knowledge model. Further, the master and the assistant divide the focus of learning where the master uses both global and local image crops, whereas the assistant specializes in global crops.

Convolution is a local operator whereas transformer is a global operator that considers all the pixels and applies attention to find the important features. It is known that effective receptive fields of lower layers for ViTs are much larger than ResNets, which helps them to incorporate more global information than CNNs [13, 14]. Hence we employ ViT for both the master and the assistant blocks.

Several methods apply various post-processing techniques like Average Query Expansion (AQE) [15], heat diffusion [16], offline diffusion [17] and CVNet-Rerank [10] to obtain the final solution by generating a ranked list of similar images. Thus we classify the image retrieval models in two categories: model without post-processing techniques and model with post-processing techniques. Some models [6–9] also include post-processing as part of their end-to-end pipeline, thus we classify them in the later category.

Contributions (1) We introduce the MABNet framework, which exploits both the supervised and the self-supervised learning paradigm. (2) We propose the use of the master and the assistant block with the master focusing on general knowledge learning and the assistant helping the master to refine its knowledge base through weight transfer. (3) We categorize all the image retrieval models in either with or without post-processing approaches. We train MABNet on the GLDv2-train-clean (GLDv2-TC) [18] dataset and demonstrate its efficacy on popular benchmark datasets, namely Oxford (Ox5k and Ox105k) [19], Paris (P6k and P106k) [20], Revisited Oxford (ROx(M) and ROx(H)) [21] and Revisited Paris (RPar(M) and RPar(H)) [21] in both the categories.

*FUNDING BY VIRTUALSTAIN PROJECT (CRISTIN ID 2061348)
†Equal contribution

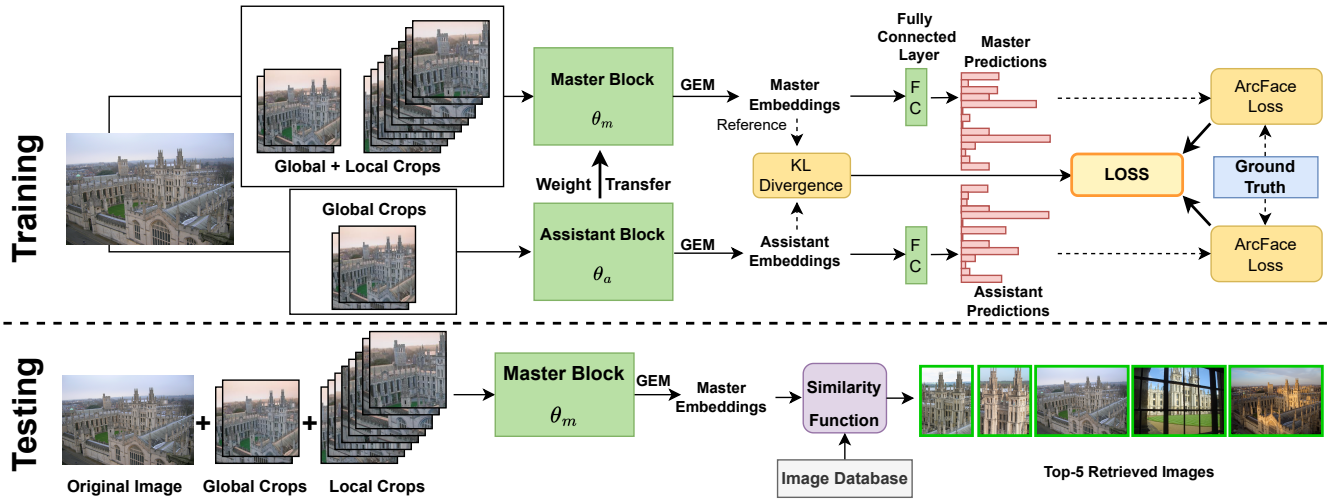

Fig. 1. *Training architecture* (*Upper part*): The architecture of the master assistant buddy network (MABNet). Global crops are fed to assistant block while both local and global crops are used for master block to generate assistant and master embedding, respectively, from Generalized Mean Pooling (GEM) layer. KL divergence loss is calculated over these embeddings. The embedding are then passed through a fully connected layer to get predictions, over which ArcFace loss is applied. After each epoch, the weights of master block are updated by weight transfer. *Testing architecture* (*Lower part*): The predictions at test time are only performed via the master block. The original input image and its global and local crops are passed to the trained master block to get the embedding from the GEM layer. The concatenation of these embeddings are then compared with the embedding of the images from database with a similarity function (any post-processing method) to retrieve k best images.

2. METHOD

The proposed MABNet (see Fig. 1) consists of two blocks: the master block and the assistant block. Both the blocks have the backbone of ViT (can be any architecture like ResNet) denoted by V but with different parameters θ_m and θ_a respectively. The role of master is to perform image retrieval while focusing on local spatial context but retaining global context as well and the role of assistant is to assist the master block by providing global spatial information through weight sharing, thereby enriching the representations learned by the master.

In MABNet, instead of passing the original input image x to the network directly, we create multiple crops of the input image using the multi-crop method. We refer to the crops covering less than 50% of the original image as local crops (denoted by C_L), and all other crops as global crops (C_G). The master receives all the crops, whereas only the global crops are passed through the assistant. For an input image x, we get two embedding E_m and E_a from the GEM layer corresponding to the master and assistant blocks, respectively.

The KL divergence loss is calculated between the embeddings E_m and E_a. A fully-connected linear layer transforms the embeddings to the corresponding output labels. We employ ArcFace to get the supervised loss of each block. The gradient of the each block with respect to the ArcFace loss flows independently and with respect to KL loss flows collectively. Further, we employ a weight sharing technique from assistant to master, to update the master block's knowledge model. The final prediction is the output of the master since it captures more generalized information than the assistant.

Weight Transfer Both the blocks learn their weights by back-propagation via the two losses, self-supervised KL divergence loss and supervised ArcFace loss. In addition to gradient descent, the weights of the master block are updated by the weighted average of master and assistant parameters as $\theta_m = \lambda\theta_m + (1-\lambda)\theta_a$, where λ is the weight transfer parameter. The value of λ is determined by grid approach and is in the range $0 \leq \lambda \leq 1$.

Testing Phase For inference, we feed forward the input image and all its crops, i.e., C_G and C_L through the master block to get the embeddings coming from the GEM layer. Therefore, we have C_G+C_L+1 embeddings. We concatenate all of them. The embedding size for each image is 512. Then we train a simple KNN over these embeddings and get the nearest neighbours of the input image. Alternatively, we can use any post-processing techniques to get the final solution by generating a ranked list of similar images in the dataset.

Model Working Figure 2 shows an example of the progressive learning of master and assistant block. The master embedding are more general and spread initially since it sees both the global and local crops. The assistant embedding is also spread but only to a part of the image depending on the global crops passed to it. But the assistant block compares its notes with the master block using self-supervision and hence learns the localised information. This can be seen by the heat maps in the later epochs as it becomes more focused at one

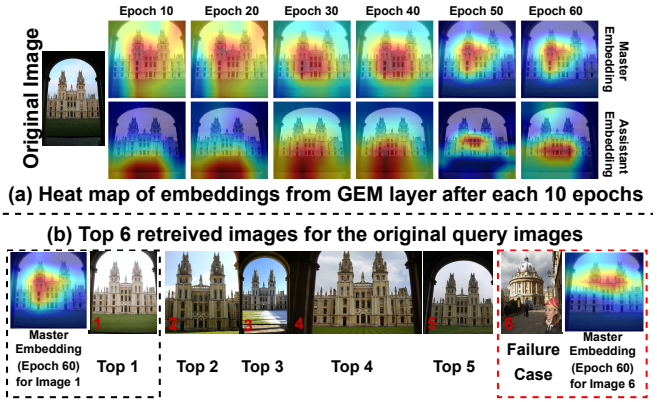

Fig. 2. (a) The superimposed heat map of the the original image after every 10 epochs from both the master and the assistant embedding is generated using Grad-CAM till 60 epochs. (b) Top 5 retrieved image for the original query image. We also show the top 6 retrieved image which is a failure case. The left and right heat map is from the master embedding for the top 1 retrieved image and the failure case, respectively. This result is on Ox105k dataset without post-processing.

particular area. The master block in turn gets this localised information from assistant block in the form of the weight transfer and thus becomes more focused in the later epoch forming better decision boundary. Example of retrieved images is shown in the lower panel of Fig. 1 and Fig. 2(b).

Setup We take 2 global and 8 local crops of random size. Random rescale is performed within the range [0.5, 1] and [0.05, 0.5] on global and local crops, respectively. The global and local crops are resized to 224×224 and 96×96, respectively. We use ViT-small pre-trained on ImageNet as the backbone. Input patch size of 16×16 is flattened and projected to M lower dimensional linear vectors. The location prior is provided by adding trainable position encoding to the linear embedding before feeding them to the transformer encoder. The model is trained with 8×V100s for 100 epochs with a batch size of 256. We use AdamW optimizer with a learning rate of 0.0005 and a weight decay of 10^{-5}. Furthermore, we use a linear warm-up for the first 10 epochs, after which the learning rate follows a cosine schedule. We use mixed-precision training to speed up computation. λ is set as 0.5. We employ the standard mean average precision (mAP) metric to report our accuracy. The training time on GLDv2-TC [18] is 2.5 days. The number of flops is 9.2 GFlops. The inference time for all the concatenated emebeddings of one image is 132 ms and for only the original image is 11 ms.

3. EXPERIMENTS

3.1. Without Post Processing

Oxford and Paris datasets The upper part of Table 1 presents our results on the Oxford and Paris data. Our model surpasses the reported SOTA performances by a significant margin (6.1% and 7.3% better on the Ox105k and P106k datasets, respectively). The mAP score of our model is highest in Ox5k and P6k too. ***Revisited datasets*** MABNet outperforms other models by a margin of 15.1% on ROx(M), 13.3% on ROx(H), 3% on RPar(M) and 8.4% on RPar(H) (see Table 1).

Table 1. mAP scores of various methods without post-processing on Oxford, Paris and their revisited datasets. MABNet-R50 represent MABNet with ResNet-50 backbone.

Model	Ox5k	Ox105k	P6k	P106k
R-MAC [1]	66.9	61.6	83.0	75.7
siaMAC [2]	80.0	75.1	82.9	75.3
DIR [3]	83.1	78.6	87.1	79.7
DELF [4]	83.8	82.6	85.0	81.7
Deep Conv [5]	83.8	80.6	88.3	83.1
MABNet	**91.5**	**88.7**	**94.2**	**90.4**
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
DINO [13]	51.5	24.3	75.3	51.6
IRT(R) [14]	55.1	28.3	72.7	49.6
Listwise [11]	67.5	42.8	80.1	60.5
SOLAR [12]	69.9	47.9	81.6	64.5
MABNet	**85.0**	**61.2**	**84.6**	**72.9**
MABNet-R50	80.3	59.2	82.0	67.4

3.2. With Post-processing

Oxford and Paris datasets Cyan colored rows in Table 2 shows that the post-processing techniques improve the performance of MABNet. Offline diffusion provides the best improvement (3.9−7.6%). MABNet is superior to other methods when applied the same post-processing methods. ***Revisited datasets*** All the methods mentioned in the lower half of Table 2 either uses the post-processing after the inference or they include post-processing technique as part of their model. Hence, we place them in the post-processing category. As seen from Table 2, MABNet with post-processing outperforms other methods in most of the cases.

3.3. Ablation Studies

We conducted a variety of ablation studies for MABNet on Ox5K dataset w/o post-processing (unless otherwise stated).
(i) ***Weight Transfer*** We considered training without weight transfer from the assistant to the master. The score dropped from 91.5% to 81.9%, which indicates its importance. We also investigated the effect of changing the direction of weight transfer, i.e., the assistant gets updated using weight transfer. This resulted in the drop of score to 80.5%. We hypothesize that the weight transfer of master instead of assistant is more effective since addition of global-only spatial information from assistant to the local-global contextual embedding

Table 2. mAP scores of various methods with different post-processing techniques on Oxford, Paris and their revisited datasets. Legend: PP - PostProcessing, Q - AQE, AQ - AML + AQE, DQ - DIR + AQE, WQR - HeW + AQE + HeR, O - Offline Diffusion, CV - CVNet.

Model + PP	Ox5k	Ox105k	P6k	P106k
MABNet	91.5	88.7	94.2	90.4
siaMAC + Q [2]	85.4	82.3	87.0	79.6
DIR + Q [3]	89.0	87.8	93.8	90.5
MABNet + Q	92.3	89.6	95.2	91.1
R-MAC + AQ [1]	77.3	73.2	86.5	79.8
DELF + DQ [4]	90.0	88.5	95.7	92.8
siaMAC+WQR [16]	92.0	90.3	94.3	90.2
MABNet + WQR	95.7	92.9	96.5	95.6
R-MAC + O [17]	96.2	95.2	97.8	96.2
MABNet + O	**97.2**	**96.3**	**98.1**	**96.8**
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
DELG [6]	81.2	64.0	87.2	72.8
DOLG [7]	81.5	61.1	91.0	80.3
Swin-T-DALG [8]	78.7	54.7	88.2	76.3
Swin-S-DALG [8]	79.9	57.5	90.4	79.0
GeM (Baseline) [9]	83.0	65.5	90.2	80.7
GeM-Local Match [9]	85.9	71.2	92.0	**83.7**
CVNet-Rerank [10]	87.2	75.9	91.2	81.1
MABNet + Q	87.1	63.8	86.4	73.5
MABNet + WQR	88.4	65.4	88.3	75.8
MABNet + O	**89.3**	66.2	88.9	78.2
MABNet + CV	87.1	**76.5**	**92.3**	82.7

of the master enriches it into a more generalized block.

(ii) ***KL Divergence Loss*** We consider the effect of dropping KL loss while training our model. The performance deteriorates from 91.3% to 83.3%, indicating that self-supervision is a critical determinant of MABNet. We also consider reversing the reference for KL loss, i.e., setting assistant embedding as reference. The model performs poorer than MABNet (by 5.8%), indicating the importance of master as reference.

(iii) ***Reversing the knowledge flow*** We considered changing the direction of both the weight transfer and KL divergence at the same time. The performance of MABNet drops by 11% asserting the importance of the chosen knowledge flow.

(iv) ***Assistant block*** We train our method with only the master block, removing the assistant block and all its dependent connections like weight transfer, KL and ArcFace loss. For fair comparison, we consider all the crops and the original image. The score here drops to 80.2% from 91.5% for Ox5k, 77.6% from 85.0% for ROx(M) and 49.3% from 61.2% for ROx(H), indicating the importance of the assistant block.

(v) ***ViT backbone*** We use ViT as the backbone due to its advantages (discussed in section 1). We replaced ViT with ResNet-50 and followed the same training procedure. Table 1 shows that ViT gives significantly better performance than ResNet-50. Nonetheless, MABNet with ResNet-50 still performs better than other methods.

Table 3. mAP scores of MABNet(C) without post-processing on Oxford, Paris and their revisited datasets.

Model	Ox5k	Ox105k	P6k	P106k
MABNet(C)	92.5	89.6	95.3	92.1
MABNet(C) + Q	93.6	92.4	95.9	94.1
MABNet(C) + WQR	97.1	93.8	98.1	96.4
MABNet(C) + O	98.3	97.8	98.3	97.5
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
MABNet(C)	85.8	62.1	85.7	73.8
MABNet(C) + Q	88.1	62.8	86.2	74.7
MABNet(C) + WQR	89.7	65.8	87.6	76.9
MABNet(C) + O	90.2	67.5	89.8	79.4

(vi) ***Concatenated features*** We use original image, 2 global and 8 local crops for inference. The mAP score of MABNet on Ox5k is 91.5%. Passing only the original image as input for inference gives 90.2% which is still superior than the SOTA. The mAP scores were 85.7% when only 2 global crops were passed and 85.4% with only 8 local crops. Increasing the concatenated features with the original image, 4 global and 12 local crops gave mAP 91.7% which is not a significant boost from 91.5%. We conclude that the use of original image, 2 global and 8 local crops gives better balance between complexity and performance. Still, to decrease the inference time, one may use only original image giving satisfactory result.

(vii) ***Ensemble of two blocks*** MABNet uses only the master block to generate the final output. Here, we extend MABNet to MABNet(C), where we predict by concatenating the embeddings of the master and the assistant block. It is seen, MABNet(C) performs better in all the situations (see Table 3). MABNet is two times faster than the MABNet(C) at inference time because it has half the number of parameters. The choice of MABNet and MABNet(C) is based on the trade-off between the inference time and the improved accuracy.

4. CONCLUSION

We presented MABNet, a new approach to employ both the supervised and self-supervised learning paradigm. Our ablation study clearly shows the importance of the buddy model, guidance of self-supervised learning by the master, and knowledge propagation from assistant to master. We showcase the efficacy of MABNet on Oxford, Paris and their revisited datasets with and without post-processing. The performance can be further improved by an ensemble of two blocks during prediction, but it doubles the inference time. The use of supervision and self-supervision collectively, can be explored for other computer vision tasks in the future. The code is available at **https://github.com/Rohit102497/MABNet**.

5. REFERENCES

[1] Giorgos Tolias, Ronan Sicre, and Hervé Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *International Conference on Learning Representations*, 2016.

[2] Filip Radenović, Giorgos Tolias, and Ondřej Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–20.

[3] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus, "Deep image retrieval: Learning global representations for image search," in *European Conference on Computer Vision*, 2016, pp. 241–257.

[4] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han, "Large-scale image retrieval with attentive deep local features," in *IEEE International Conference on Computer Vision*, 2017, pp. 3456–3465.

[5] Tuan Hoang, Thanh-Toan Do, Dang-Khoa Le Tan, and Ngai-Man Cheung, "Selective deep convolutional features for image retrieval," in *ACM International Conference on Multimedia*, 2017, pp. 1600–1608.

[6] Bingyi Cao, Andre Araujo, and Jack Sim, "Unifying deep local and global features for image search," in *European Conference on Computer Vision*. Springer, 2020, pp. 726–743.

[7] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang, "Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11772–11781.

[8] Yuxin Song, Ruolin Zhu, Min Yang, and Dongliang He, "Dalg: Deep attentive local and global modeling for image retrieval," *arXiv preprint arXiv:2207.00287*, 2022.

[9] Zechao Hu and Adrian G Bors, "Expressive local feature match for image search," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 1386–1392.

[10] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim, "Correlation verification for image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5374–5384.

[11] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5107–5116.

[12] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk, "Solar: second-order loss and attention for image retrieval," in *European Conference on Computer Vision*. Springer, 2020, pp. 253–270.

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging properties in self-supervised vision transformers," in *IEEE International Conference on Computer Vision*, 2021, pp. 9650–9660.

[14] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou, "Training vision transformers for image retrieval," 2021.

[15] Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas, "Total recall ii: Query expansion revisited," in *CVPR 2011*. IEEE, 2011, pp. 889–896.

[16] Shanmin Pang, Jin Ma, Jianru Xue, Jihua Zhu, and Vicente Ordonez, "Deep feature aggregation and image reranking with heat diffusion for image retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1513–1523, 2018.

[17] Fan Yang, Ryota Hinami, Yusuke Matsui, Steven Ly, and Shin'ichi Satoh, "Efficient image retrieval via decoupling diffusion into online and offline processing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 9087–9094.

[18] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2575–2584.

[19] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[20] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[21] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5706–5715.