DOT-PRODUCT BASED GLOBAL AND LOCAL FEATURE FUSION FOR IMAGE SEARCH

Zechao Hu and Adrian G. Bors

Department of Computer Science, University of York, YO10 5GH, UK

ABSTRACT

Content-based image retrieval (CBIR) consists in searching the most similar images to the query content from a given pool of images or database. Existing works' success relies on taking advantage of both local and global feature information leading to better retrieval performance than when using either of these. Lately, CBIR area has been dominated by the two-stage image retrieval framework which utilizes global features to get initial retrieval results, while using local features for re-ranking in a second stage. In this study, instead of utilizing local and global features separately during two stages, we propose to use a dot-product based local and global (DPLG) feature fusion module leading to a comprehensive global feature descriptor. The proposed fusion module is jointly end-to-end trained within the convolution backbone structure. According to the experimental results, the proposed module achieves new state-of-the-art results on some benchmark datasets.

***Index Terms*—** Content based image retrieval, Local and global features, Dot-product attention.

1. INTRODUCTION

Content-based image retrieval (CBIR) is a classic computer vision task, which received increasing attention during the last 30 years. Due to the complexity and variability of image content, the main challenge is represented by the image feature extraction and the ability to yield a compact image representation. Early conventional image feature extraction approaches rely on low-level feature information and hand-crafted extractors, which cannot bridge the gap between low-level feature representation and high-level semantic meaning.

The success of deep convolution neural networks revolutionized image feature extraction. Generally, there are two types of Convolution Neural Networks (CNN) based feature extraction methods for the CBIR task: local and global feature methods. Local feature methods preserve the correspondence between each location of the convolution feature tensor and a region from the input image. The local features would either be aggregated into a compact feature vector by a separate aggregation method [1] or directly used for evaluating the similarity in a many-to-many manner [2]. Global feature methods aim to extract a compact global feature vector from the input image by employing a single forward data processing through the model [3, 4, 5].

Generally speaking, global features are robust to viewpoint and illumination changes while local features represent low level information such as texture and contrast. To benefit from both worlds, an effective two-stage framework has been proposed recently [6, 7]. During the first initial retrieval stage, global features are utilized to get initial retrieval results. At the second, re-ranking stage, local feature vectors are used to perform nearest neighbor search and spatial verification for re-ranking, leading to an improved final retrieval performance. Although it achieves decent retrieval performance even for some hard cases, the two-stage framework still suffers from two important problems: first, the second re-ranking stage relies on a large number of local features. Even when these local features have been compressed by dimension reduction and binary encoding, it is still expansive to perform spatial verification, like for example using the classic RANSAC algorithm [8]. This is also the reason why the second stage re-ranking is strictly limited to the top 100 initial retrieval results [6, 7]. In other words, if the global features at the first stage could not rank wanted images within top 100, these images actually could not benefit from the local features at all. Second, dividing the retrieval procedure into separate two stages means that errors will be passed over between them and accumulate, which would have a negative impact on the model performance [9].

In this work, we abandon the two-stage framework and explore building a more comprehensive global image representation by fusing global and local feature information through an end-to-end trainable fusion module. The proposed fusion module is designed based on the dot-product attention mechanism [10] which is also the core component of many popular deep learning models, such as the Self-Attention mechanism [10] or the Visual Transformer [11]. We notice that the recent CBIR work DOLG [9] share a similar motivation with our work but our method shows better retrieval results under the exact same setting.

The contributions of the paper are: 1) we propose an effective dot-product attention based module for global and local feature fusion. 2) we show that the dot-product attention implicitly serves as spatial attention for local feature re-weighting. 3) The global feature model performance is greatly improved, in challenging examples, when embedding the proposed fusion module, to reach new state of the art performance on various benchmark datasets.

2. RELATED WORK

Local feature methods. Many local feature aggregation approaches using deep CNNs, rely on conventional approaches such as the Vector of Locally Aggregated Descriptors (VLAD) [12]. NetVLAD [13] modifies VLAD as an end-to-end trainable layer at the tail of a CNN structure. The experimental results show that the trainable VLAD outperforms the local feature fusion methods which are not based on deep learning. HOW [2] employs the Aggregated Selective Match Kernel (ASMK) [14] to directly perform many-to-many local feature matching with the features yield by a CNN, reaching good balance between performance and computation cost. The DEep Local Feature (DELF) [6] is a representative two-stage local feature model utilizing local features to re-rank the initial retrieval results. It implements a score function with two processing layers on top of the final convolution layer for relevant local feature selection. During the initial retrieval stage, the compact global feature vector is built by a weighted sum of selected local features. During the re-ranking stage, after dimension reduction, geometry verification is performed with these local features to get the final retrieval result. DEep Local and Global features (DELG) model [7], was derived from DELF, by unifying the training procedures of global and local features into a single pipeline.

Global feature methods. The earliest deep CNN-based global feature method for CBIR task can be tracked back to the Neural Code model [15] where a feature vector is extracted by a fully connected layer. After Razavian *et al.* [16], proved that spatial pooling is better for instance feature extraction, various studies proposed sum-pooling [3], max-pooling [4] and generalized mean pooling [5]. Attention mechanisms have been embedded into the global feature extraction pipeline for better global image representation as in the Weighted Generalized Mean pooling (WGeM) [17] which applies a trainable spatial weighting module by adding an extra convolutional layer, or in the conditional attention mechanism [18]. WGeM can effectively localize the objects of interest while ignoring redundant regions. The Second-Order Loss and Attention for image Retrieval (SOLAR) [19] explored the co-relations at each location in the CNN feature map using the second-order spatial information.

The Deep Orthogonal Local and Global (DOLG) [9] proposes a more comprehensive global feature extraction pipeline, in which an Orthogonal Fusion module complements the global feature vector with critical local feature information leading to the current state-of-the-art results for CBIR. The orthogonal fusion module in DOLG computes residual vectors between the global feature vector and that corresponding to each location. The resulting residual vectors are summed and serve as complementary to the global feature vector. We can see that the strategy of complementary local feature information extraction from the local feature tensor is pre-fixed by the orthogonal design. In our proposed approach, we make the fusion strategy end-to-end trainable by the proposed dot-product based fusion module.

3. DOT-PRODUCT FEATURE FUSION FOR CBIR

3.1. The processing pipeline

The proposed CBIR methodology is illustrated in Figure 1. We consider Resnet50 [20] as the backbone network. In Resnet50 the initial processing blocks Res1-Res5 indicate 5 residual convolution blocks. Unlike in most existing global feature methods that would only use the output of the final convolution layer for feature extraction, we keep both feature tensors $\mathbf{X}_4 \in I\!R^{H_4 \times W_4 \times D_4}$ and $\mathbf{X}_5 \in I\!R^{H_5 \times W_5 \times D_5}$ ($H_4 = H_5 \times 2, W_4 = W_5 \times 2, D_4 = 1024, D_5 = 2048$) which are the image representation outputs of Res4 and Res5.

The global feature tensor $\mathbf{X}_5$ is fed into the GeM pooling layer, resulting in $\mathbf{V}_g \in I\!R^{1 \times D_5}$, which is defined by :

$$\mathbf{V}_g = \left(\frac{1}{L_5} \sum_{l}^{L} \mathbf{X}_{5,l}{}^{p} \right)^{\frac{1}{p}}, \qquad (1)$$

where $L_5 = H_5 \times W_5$, $l = 1, \ldots, L_5$ indicates entries on $\mathbf{X}_{5,l}$. The power coefficient is set as $p = 3$, as in other works. Dimension reduction is performed on $\mathbf{V}_g$ by a convolution layer with kernel size 1×1, resulting in the global feature vector $\mathbf{V}'_g \in I\!R^{1 \times D_4}$.

The local feature tensor $\mathbf{X}_4$ is first fed into the Receptive field block [21] for feature refinement and then into the Dot-product attention module along with the global feature vector $\mathbf{V}'_g$, resulting in a compact local feature information feature vector $\mathbf{V}_l \in I\!R^{1 \times D_4}$. Local feature information $\mathbf{V}_l$ and global feature $\mathbf{V}'_g$ are then element-wise added and fed into a fully connected layer (FC) to produce the final fused global feature vector, $\mathbf{F} \in I\!R^{1 \times D}$. The core components of the proposed model pipeline are the Receptive Field Block (RFB) and the Dot-product fusion module.

3.2. Pyramid features by the Receptive field block (RFB)

RFB, adapted from [21], represents a pyramid feature extraction block which handles different target object scales. RFB consists of several convolution layer branches, where each branch contains 2 convolution layers. The first convolution layer has various kernel sizes while the second varies the dilation rate. This design enables the feature extraction pipeline not only just for multiple receptive fields but also for different resolutions from previous convolution layers. This attribute enables it to catch features at different scales while each entry is more distinctive [21]. An additional global GeM pooling branch is added at the bottom of RFB in Figure 1. The output of 3 convolution layer branches along with the GeM pooling branch are concatenated[1] then fed into a convolution layer to output the final refined local feature tensor $\mathbf{X}'_4$.

[1] The feature vector output by GeM pooling is expanded to the same size as the other 3 convolution layers' outputs

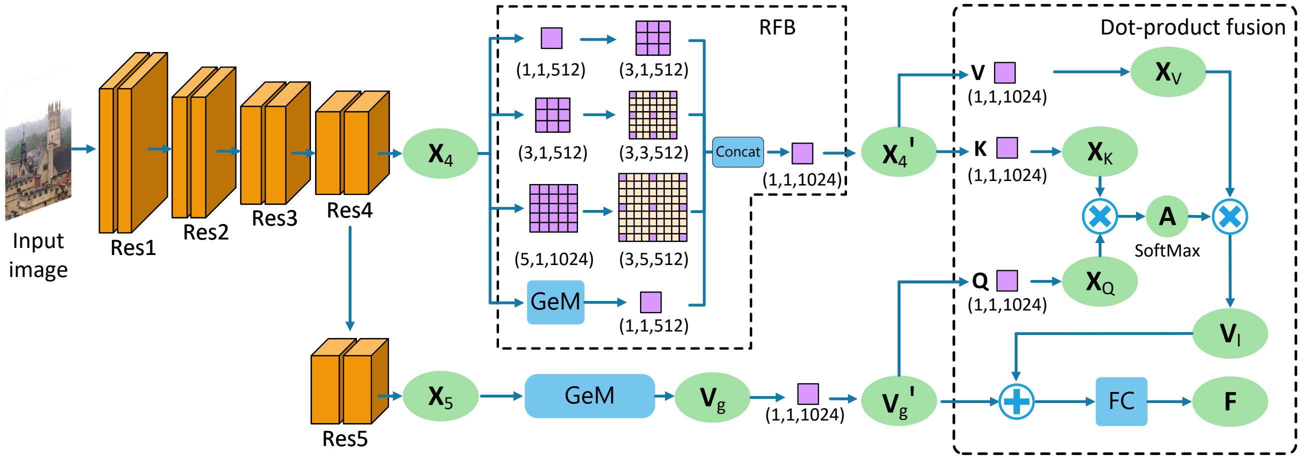

Fig. 1. Illustration of the DPLG model structure. Purple grids indicate convolution layers of different kernel sizes and dilations, where (k, r, d) represent kernel size, dilation rate and output channel (dimension), respectively.

3.3. The Dot-product fusion module

The dot-product fusion module, shown in the right part of Figure 1, takes the local feature tensor $\mathbf{X}_4'$ and the global vector $\mathbf{V}_g'$ as inputs. First, the local feature tensor $\mathbf{X}_4'$ is mapped and reshaped into $\mathbf{X}_K, \mathbf{X}_V \in I\!R^{(H_4 \times W_4) \times D_4}$ separately by convolution layers K and V, while the global feature vector $\mathbf{V}_g'$ is mapped to $\mathbf{X}_Q \in I\!R^{1 \times D_4}$ by the convolution layer Q[2]. The dot-product weight matrix $\mathbf{A} \in I\!R^{1 \times (H_4 \times W_4)}$ is :

$$\mathbf{A} = \text{softmax}\left((\mathbf{X}_Q \mathbf{X}_K^{\mathsf{T}}) / \sqrt{D}_4 \right). \qquad (2)$$

$\mathbf{A}$ is a sequence of weights with respect to all locations on the local feature tensor $\mathbf{X}_4'$. The dot-product attention weighted local feature information vector $\mathbf{V}_l \in I\!R^{1 \times D_4}$ is defined by :

$$\mathbf{V}_l = \mathbf{A}\mathbf{X}_V. \qquad (3)$$

Finally, feature vectors $\mathbf{V}_g'$ and $\mathbf{V}_l$ are element-wise added and fed into a fully connected layer generating the final global descriptor for the input image.

What the dot-product fusion aims for? Intuitively speaking, the feature tensor output by a shallow residual block, like $\mathbf{X}_4$, contains more localized and relatively low-level feature information as it was processed by fewer convolution and down-sampling layers. On the contrary, the feature output by the deep residual block, like $\mathbf{X}_5$, has higher-level semantic meaning and wider receptive field due to the additional convolution layer processing. The purpose of the dot-product fusion is to enable the feature extraction pipeline with the ability to dynamically extract additional complementary local information from the shallow layer output $\mathbf{X}_4$, representing the local information, and with respect to the global feature vector $\mathbf{V}_g$ and then fuse them together to build a more comprehensive global descriptor $\mathbf{F} \in I\!R^{1 \times D}$. Compared with the orthogonal fusion, our fusion strategy is not pre-fixed and it is automatically learned by the proposed dot-product fusion module during the training stage.

3.4. Implementation details

For the model training, following the practice from DELG [7], we also consider image-level class labels and the ArcFace margin loss [24], defined by:

$$L(\widehat{\mathbf{V}}_g, \mathbf{y}) = -\log \frac{\exp(\gamma \times \text{AF}(\widehat{\mathbf{V}}_g \widehat{\mathbf{w}}_i^{\mathsf{T}}, y_i))}{\sum_{j=1}^{N_c} \exp(\gamma \times \text{AF}(\widehat{\mathbf{V}}_g \widehat{\mathbf{w}}_j^{\mathsf{T}}, y_j))}, \qquad (4)$$

where $\widehat{\mathbf{F}}$ represents the L2 normalized global feature vector $\mathbf{F}$ while $\widehat{\mathbf{w}}_i$ is the trainable proxy feature vector $\mathbf{w}_i$ for class i from the ArcFace weight matrix $\mathcal{W} \in I\!R^{N_c \times D}$, N_c is the number of classes in the training dataset. $\mathbf{y}$ is the one-hot class label vector. $\text{AF}(u, y)$ is the ArcFace-adjusted cosine similarity [7]:

$$\text{AF}(u, y) = \begin{cases} \cos(\arccos(u) + m), & \text{if } y = 1 \\ u, & \text{if } y = 0 \end{cases}. \qquad (5)$$

We set the ArcFace margin $m = 0.15$ and temperature $\gamma = 30$. The model is trained on the GLDv2 dataset [25], using the SGD optimizer with cosine learning rate decay strategy, initial learning rate 0.03. The model is trained for no more than 100 epochs. The retrieval performance is evaluated on ROxf/RParis datasets [26]. During the evaluation stage, a multi-scale feature extraction scheme [5] is applied with 5 scales: $\{\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$. The feature dimension output by the final fully connected layer is set to $D = 512$.

4. EXPERIMENTAL RESULTS

Retrieval results of the proposed model and comparisons with the state-of-the-art are provided in Table 1. We mainly focus on comparison to the current state of the works HOW

[2] For expression consistency, we implement Q as a convolution layer with kernel size 1 while $\mathbf{V}_g'$ is reshaped into $I\!R^{1 \times D_4 \times 1 \times 1}$ to meet the input shape requirement for the convolution layer

Method	*Medium (%)*				*Hard (%)*			
	ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
(A) Local features + re-ranking								
DELF-ASMK*+SP [22]	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
DELF-D2R-R-ASMK*+SP [22]	76.0	64.0	80.2	59.7	52.4	38.1	58.6	29.4
R50-HOW [2]	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
(A) Global features + re-ranking								
R101-R-MAC [23]	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
R101-GeM (GLD) [5]	67.3	49.5	80.6	57.3	44.3	25.7	61.5	29.8
R101-SOLAR [19]	69.9	53.5	81.6	59.2	47.9	29.9	64.5	33.4
R50-DELG (GLDv2) [7, 9]	77.5	74.8	87.9	77.3	54.8	50.4	73.8	61.0
R50-DELG (GLDv2) + SP [7, 9]	79.1	75.9	88.8	77.7	58.4	52.4	76.2	61.6
R50-DOLG [9]	80.5	76.6	89.8	80.8	58.8	52.2	77.7	**62.8**
(C) Our method								
R50-DPLG	**81.1**	**77.2**	**90.0**	**81.7**	**60.2**	**53.1**	**78.4**	62.0

Table 1. Mean average precision (mAP) on ROxf/RPar datasets (with 1M distractor set), considering *Medium* and *Hard* evaluation protocols. All methods have Resnet50 as their backbone for fair comparison. "SP" is Spatial verification.

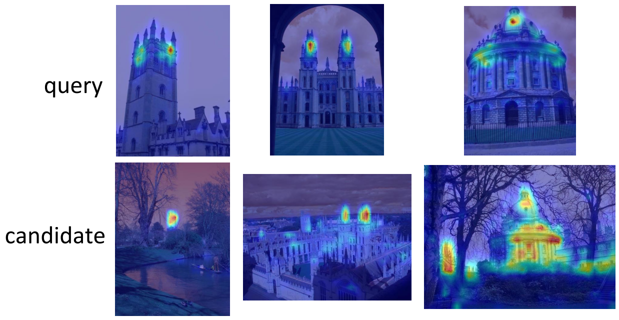

Fig. 2. Visualization of the dot-product attention matrix $\mathbf{A}$.

[2], DELG [7] and DOLG [9] considering Resnet50 as the backbone structure. The results for DELG come from the report in DOLG paper which re-trained DELG on the GLDv2 dataset by the same training setting for 100 epochs for fair comparison. Our model outperforms all other methods except for RPar+1M with the *Hard* evaluation protocol. However, our model surpasses DOLG on the *Hard* protocol of ROxf (ROxf+1M) by 1.7%(0.9%), so our model still performs overall better than DOLG.

We visualize the attention $\mathbf{A}$ from Eq. (2) for 3 pairs of matching images in Fig. 2. We observe that the dot-product attention mainly focuses on the relevant objects' representative parts such as the building's tower in the middle image while the background information is removed from the field of interest. The dot-product mechanism serves as a spatial attention mechanism to pick out only important local features for global and local feature fusion. However, sometimes the dot-product attention could uniformly highlight all landmark-like objects like the remote building from the left side is also highlighted in the third image from the bottom row of Fig. 2.

Table 2 contains the ablation study for the impact of each component from the proposed processing pipeline. When not using the RFB (3rd row), each location on the local feature tensor $\mathbf{X}_4'$ would not be able to represent the object and the model performance is decreased. When not using the Dot-product fusion (2nd row) but naïvely pooling the local feature tensor $\mathbf{X}_4'$ and adding it to the global feature vector $\mathbf{V}_g'$ would just uniformly embed all relatively low-level local feature information into the global feature vector, making no positive contribution to the model performance.

RFB	Dot-product	*Medium* (%)		*Hard* (%)	
		ROxf	RPar	ROxf	RPar
✗	✗	78.3	88.0	56.1	73.9
✓	✗	78.5	87.8	55.5	74.0
✗	✓	80.8	89.4	59.3	77.5
✓	✓	81.1	90.0	60.2	78.4

Table 2. mAP results on ROxf and RPar datasets, where without RFB means that we replace RFB with a 1×1 kernel size convolution layer. Without Dot-product means the local feature tensor $\mathbf{X}_4'$ is the global GeM pooled then element-wise added with $\mathbf{V}_g'$. Without both RFB and Dot-product means the model is the same to original GeM from [5].

5. CONCLUSION

In this paper, we propose an effective dot-product fused local and global feature fusion module for content-based image retrieval. Unlike existing feature fusion mechanisms which use a fixed strategy to extract complementary local information from the local feature tensor. The proposed dot-product fusion module is automatically learned during the training. We also demonstrate that the dot-product attention also implicitly learns a spatial weighting mechanism which is good at masking out irrelevant information such as sky or grass from the background. When including the fusion module, the proposed model globally outperforms current state of the art works on common benchmark datasets.

6. REFERENCES

[1] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. of CVPR-workshops, vol. LNCS 9913*, 2015, pp. 685–701.

[2] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum, "Learning and aggregating deep local descriptors for instance-level recognition," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12346*, 2020, pp. 460–477.

[3] Artem Babenko and Victor Lempitsky, "Aggregating local deep features for image retrieval," in *Proc. IEEE Int. Conf. on computer vision (ICCV)*, 2015, pp. 1269–1277.

[4] Giorgos Tolias, Ronan Sicre, and Hervé Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1511.05879*, 2016.

[5] Filip Radenović, Giorgos Tolias, and Ondřej Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[6] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 3456–3465.

[7] Bingyi Cao, André Araujo, and Jack Sim, "Unifying deep local and global features for image search," in *Proc. European Conf. on Computer Vision (ECCV), vol LNCS 12365*, 2020, pp. 726–743.

[8] Martin A Fischler and Robert C Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] 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 *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, October 2021, pp. 11772–11781.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in Neural Inf. Proc. Systems (NeurIPS)*, vol. 30, pp. 6000–6010, 2017.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2010.11929*, 2021.

[12] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2011.

[13] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proc. IEEE Conf. on Computer Vision and pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

[14] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou, "Image search with selective match kernels: aggregation across single and multiple images," *Int. Journal of Computer Vision*, vol. 116, no. 3, pp. 247–261, 2016.

[15] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural codes for image retrieval," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 8689*, 2014, pp. 584–599.

[16] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki, "Visual instance retrieval with deep convolutional networks," *ITE Trans. on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016.

[17] Xiaomeng Wu, Go Irie, Kaoru Hiramatsu, and Kunio Kashino, "Weighted generalized mean pooling for deep image retrieval," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2018, pp. 495–499.

[18] Zechao Hu and Adrian G. Bors, "Conditional attention for content-based image retrieval," in *Proc. British Machine Vision Conference (BMVC)*, 2020.

[19] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk, "Solar: second-order loss and attention for image retrieval," in *Proc. European Conf. on Computer Vision (ECCV), vol. LNCS 12370)*, 2020, pp. 253–270.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[21] Songtao Liu, Di Huang, and Yunhong Wang, "Receptive field block net for accurate and fast object detection," in *Proc. of the European Conference on Computer Vision (ECCV), vol. LNCS 11215*, 2018, pp. 404–419.

[22] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5109–5118.

[23] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. European Conf. on computer vision (ECCV), vol. LNCS 9910*, 2016, pp. 241–257.

[24] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[25] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2572–2581.

[26] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting Oxford and Paris: Large-Scale image retrieval benchmarking," in *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5706–5715.