https://doi.org/10.2352/ISSN.2470-1173.2019.8.IMAWM-406

Comparison of texture retrieval techniques using deep convolutional features

Augusto C. Valente; Eldorado Research Institute; Campinas, SP, Brazil
Fábio V. M. Perez; Eldorado Research Institute; Campinas, SP, Brazil
Guilherme A. S. Megeto; Eldorado Research Institute; Campinas, SP, Brazil
Marcos H. Cascone; Eldorado Research Institute; Campinas, SP, Brazil
Otavio Gomes; Eldorado Research Institute; Campinas, SP, Brazil
Thomas S. Paula; HP Inc; Porto Alegre, RS, Brazil
Qian Lin; HP Labs, HP Inc; Palo Alto, California, USA

Abstract

Considering the complexity of a multimedia society and the subjective task of describing images with words, a visual search application is a valuable tool. This work implements a Content-Based Image Retrieval (CBIR) application for texture images with the goal of comparing three deep convolutional neural networks (VGG-16, ResNet-50, and DenseNet-161), used as image descriptors by extracting global features from images. For measuring similarity among images and ranking them, we employed cosine similarity, Manhattan distance, Bray-Curtis dissimilarity, and Canberra distance. We confirm that global average pooling applied to convolutional layers provides good texture descriptors, and propose to use it when extracting features from VGG-based models. Our best result uses the average pooling layer from DenseNet-161 as a 2208*-dim feature vector along with Bray-Curtis dissimilarity. We achieved* 73.09% *mAP@1 and* 76.98% *mAP@5 on the Describable Textures Dataset (DTD) benchmark, adapted for image retrieval. Our mAP@1 result is comparable to the state-of-the-art classification accuracy (73.8%). We also investigate the impact on retrieval performance when reducing the number of feature components with PCA. We are able to compress a* 2208*-dim descriptor down to* 128 *components with a moderate 3.3 percentage points drop in mAP@1.*

Introduction

The popularity of devices capable of capturing, storing, processing and transferring multimedia data, such as image, video, and audio, is generating a growing need for information retrieval techniques in these formats, differing from traditional searches for keywords, word tags or metadata. However, the task of describing images with text is subjective and it may be very difficult to express visual information accurately. A Content-Based Image Retrieval (CBIR) system returns a list of images ordered by the visual similarity concerning a query image. As depicted in Figure 1, the CBIR system works by extracting image descriptors and ranking their visual similarities using some metric in the feature space.

Textures are characterized by repeating patterns that could provide cues of material properties. Retrieving texture images by CBIR is very desirable since words like *fibrous*, *marbled*, *dotted* or *blotchy* are not enough for describing complex patterns.

Since early 1990s several works have explored the image search task, while developing techniques for extracting features from images, for storing these features with scalability, and for comparing a query image to the database. According to Zhout et al. [1], CBIR systems face four issues: image representation, similarity measurement, search, and storage.

Deep Convolutional Neural Networks (DCNNs) represent a major breakthrough for image understanding [2]. Since then, extensive research in DCNN architectures has been conducted in order to find better image recognition algorithms. Moreover, DCNNs can act as feature extractors that are less sensitive to the semantic gap. That is, DCNNs feature vectors correlate images that are similar not only visually but also semantically. For example, images of a big black dog and a small white dog are semantically closer than pictures of a small white dog and a small white cat, since the latter pair is not from the same category.

This work provides an in-depth evaluation of the three most used DCNN architectures applied on texture retrieval: VGG [3], ResNet [4], and DenseNet [5]. We explore two critical factors in CBIR engineering:

1. Which architecture (VGG-16, ResNet-50, DenseNet-161), layer, and similarity metric (cosine similarity, Manhattan distance, Bray-Curtis dissimilarity, and Canberra distance) combination provides the best results;
2. The impact of feature vector dimensionality reduction, as a small image descriptor footprint is essential for efficient and scalable feature search and storage [6].

Related Work

Before DCNNs, image retrieval descriptors were primarily SIFT-based local features aggregated into a global descriptor [7–10]. Vector aggregation methods combine local features to encode its distribution, rather than patterns appearance order. Describing feature distribution is beneficial for retrieval tasks because it captures unique details that are relevant per instance and not per object type [11]. The Bag-of-Words (BoW) aggregated descriptor, introduced for retrieval by Sivic and Zisserman [7], is a histogram in which the bins are the the most prominent local patterns of the entire retrieval set. Later, improvements over BoW as VLAD [9] and Fisher Vectors [10] improved retrieval quality. Another retrieval technique originated in this period is the Principal Component Analysis (PCA) post-processing, used for reducing the feature vector size and for whitening [6, 9, 10, 12].

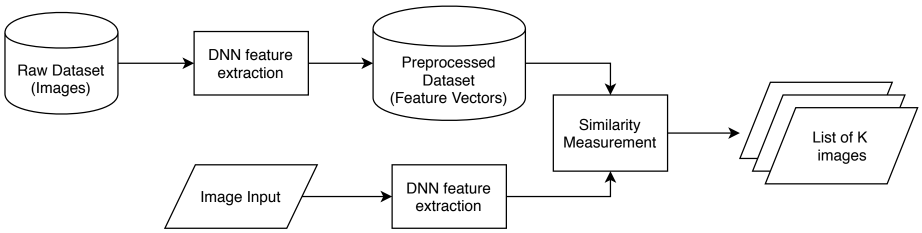

Figure 1. *Diagram of a Content-Based Image Retrieval (CBIR) system.*

After the success of DCNNs over hand-crafted features on visual recognition, many CBIR authors extracted image descriptors from pre-trained models, such as AlexNet [2, 13], VGG [12, 14–17], CaffeNet [16, 18], ResNet [19], and DenseNet [20]. Wan et al. [13] compared feature maps from fully-connected (FC) layers as global image descriptors, measuring the Euclidean distance between the query and the retrieval set. Gordo et al. [15] added a triplet-loss layer to learn a similarity metric between image pairs. Razavian et al. [12] presented a multi-scale patch pooling scheme to extract local features from the last convolutional layer, in which retrieval is done by calculating intra Euclidean distance between patches. Moreover, feature maps from convolutional layers can themselves be treated as local features [21], suitable for pooling by BoW [16], Fisher Vectors [14, 21, 22], VLAD [17] or learning [19].

The state of the art of texture representation is also based on the aggregation of hand-crafted [21, 23] or local convolutional features [14, 19, 21, 22]. According to Liu et al. [24], FC layers are an order-strict pooling mechanism, which fails to represent textures as repetitive patterns.

For ranking global features, besides calculating the Euclidean similarity [7, 12, 13, 17], other metrics include cosine [16], Hamming [8, 10], Wasserstein and Kullback Leibler [14].

In comparison with related texture analysis works [14, 19, 21–23], we opted to not include hand-crafted features methods in the pipeline to better focus on DCNN architectures. An advantage of this choice is that the overall pipeline becomes simpler and faster to execute, which is a desirable property when deploying it as an application.

Methodology

We use DCNNs to extract global descriptors from texture images, and compare them with a similarity measure between images. In this way, we are able to rank the database images based on their similarities to the query image, as shown in Figure 1. In the context of texture image retrieval, we consider a pair of images to be similar if they exhibit similar texture patterns.

We experimented with combinations of DCNN architectures, network layers, and vector similarity measures. Additionaly, we reduced the number of feature components using PCA, and analyzed if the retrieval performance was maintained.

Network architectures

We compared feature extraction capabilities for VGG-16, ResNet-50, and DenseNet-161 architectures. The global descriptor for each image is the feature vector produced by intermediary layer activations by feed-forwarding the image through the network. We used networks pre-trained on the ImageNet dataset [25] with implementation provided by the PyTorch framework [26]. We modified the architectures to extract only the desired intermediary activations, discarding all subsequent layers.

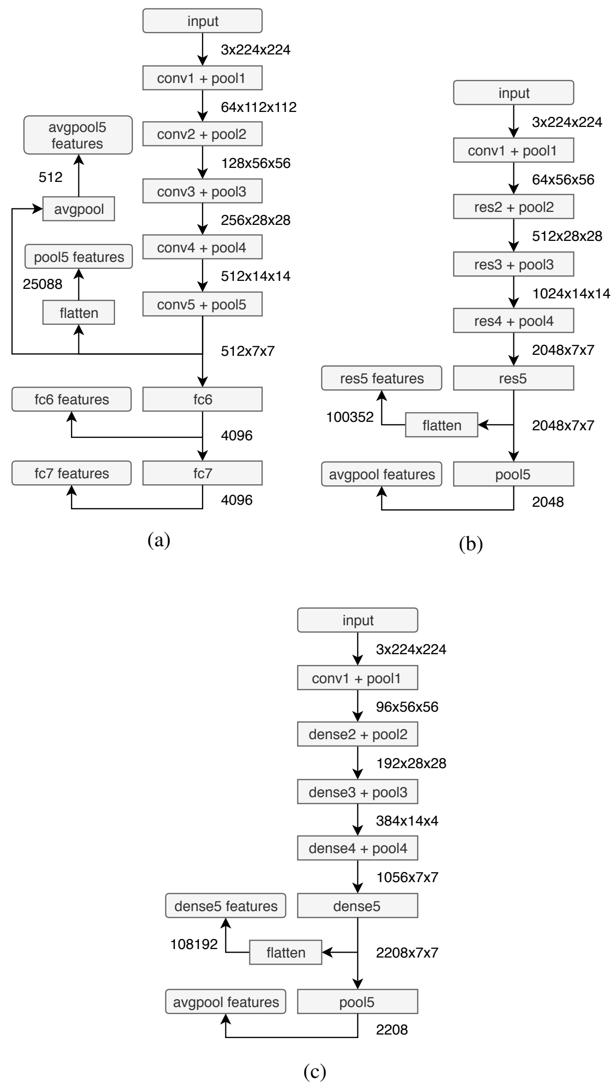

Figure 2. *(a) VGG-16, (b) ResNet-50, and (c) DenseNet-161 architectures and feature extraction locations.*

For VGG-16 (Figure 2a), we evaluated the outputs from layers *fc7* (last FC layer before classification; 4 096-dim), *fc6* (second to last FC layer; 4 096-dim), and *pool5* (flattened $512 \times 7 \times 7$ feature map from the last max pooling layer; 25 088-dim). Additionally, we experiment with global average pooling over the *pool5* convolutional feature map, obtaining the 512-dim *avgpool5* descriptor as an alternative to the 4 096-dim global descriptors from the FC layers. For ResNet-50 (Figure 2b), we evaluated the outputs from *avgpool* (global average pooling before classification output; 2 048-dim), and *res5* (flattened $2048 \times 7 \times 7$ feature map from the last residual block before *avgpool*; 100 352-dim). For DenseNet-161 (Figure 2c), we used *avgpool* (global average pooling before classification output; 2 208-dim), and *dense5* (flattened $2208 \times 7 \times 7$ feature map from the last dense block, before *avgpool*; 108 192-dim). We based the layer names on their official implementation under the Caffe framework. We extracted features from convolutional and FC layers after non-linear ReLU activation, which empirically provided better results compared to pre-activation features.

Descriptors extracted from final layers such as FC layers (*fc6* and *fc7* from VGG-16) or global average pooling (*avgpool5* from VGG-16 and *avgpool* from ResNet-50 and DenseNet-161) can be interpreted as global image descriptors, as their components are associated with the whole input image space. Global average pooling can actually be interpreted as the texture analysis method, which collects responses from filter banks, where the responses are the feature map channels, similar to the work of Andrearczyk and Whelan [27]. On the other hand, descriptors extracted from intermediary convolutional layers (*pool5* from VGG-16, *res5* from ResNet-50, and *dense5* from DenseNet-161) can be viewed as local image descriptors, which are more commonly used with an additional feature encoding step like Fisher Vectors [21]. However, we flattened these feature maps and used them directly as descriptors, for simplicity. Although this is not an optimal method, we conjectured that it is sufficient for texture comparison, which is simpler than problems such as natural scene image understanding.

Similarity measures

As aforementioned, a query image is used as the input of a CBIR system, which has the goal of retrieving images that are similar (in our case, images that have similar texture) to the input image. There are several methods to quantify similarity, and we restricted our scope to four of them: 1) cosine similarity; 2) Manhattan distance (L1-norm or city block); 3) Bray-Curtis dissimilarity (or Sorensen coefficient); and 4) Canberra distance. Manhattan distance is preferable over Euclidean distance in high dimensionality problems [28]. Canberra distance is a weighted version of the Manhattan distance and presents good results for image retrieval [29]. Cosine and Bray-Curtis measures are less sensitive to vector norm in Euclidean hyperplanes, so they can be useful when vectors share similar characteristics but differ in magnitude. Both measures have presented good results in similarity metrics works [29].

Evaluation

We followed standard image retrieval evaluation protocols and used mean average precision at K results (mAP@K) as metric, with $K = 1$ and $K = 5$.

We adapted the publicly available Describable Texture Dataset (DTD) [30], originally designed for benchmarking texture classification, for the image retrieval task. The dataset contains 5 640 texture images organized into 47 categories (120 images per category), such as *banded*, *dotted*, and *knitted*. We used the ten official train, validation, and test splits provided with the DTD dataset. We evaluated the retrieval on each fold, using the images from the test set as queries, and the retrieval database is the union of the train and validation sets. We averaged the results over all folds to report mAP values. We considered a retrieved image as successful if its category matched that of the query image. For $K = 1$, the number of successful results will either be 1 or 0, because the retrieved image category either matches or not with that of the query. For $K = 5$, the same thought applies, with successful results varying from 0 to 5.

Dimensionality Reduction

Due to the high feature dimensionality generated by DCNNs, we also experimented on how much we can reduce feature dimensionality without harming performance. To this end, we applied PCA with a target dimensionality varying from 512 to 8, and evaluated the retrieval performance at each step.

Results

Table 1 shows results for the image retrieval task on the DTD benchmark.

VGG-16 results indicate that applying global average pooling (*avgpool5*) leads to a better texture descriptor than the outputs of FC layers (*fc6*, *fc7*), pointing to the Cimpoi et al. [21] intuition that aggregation by FC layers are better at object recognition, discarding texture information, preserved by the convolutional and pooling layers.

VGG-16 has the best configuration with *avgpool5* layer and Bray-Curtis measure, achieving a mAP@1 of 68.20% and a mAP@5 of 72.68%. Additionally, for cosine or Bray-Curtis, the *pool5* descriptor has comparable results to *fc6* and *fc7* descriptors, indicating that it may be sufficient to describe simpler texture images.

DenseNet-161 with *avgpool* and Bray-Curtishas has the best results among all networks, achieving an mAP@1 of 73.09% and an mAP@5 of 76.98%. For mAP@1, this represents an increase of almost 3 percentage points (pp) over the best configuration for ResNet-50 (*avgpool* with cosine), and almost 5 pp points over the best configuration for VGG-16 (*avgpool5* with Bray-Curtis).

While VGG is widely used for information retrieval, our experiments show that ResNet-50 and DenseNet-161 outperform it. Additionally, these networks have considerably fewer parameters: 23M for ResNet-50 and 26M for DenseNet-161, versus 134M for VGG (discarding the last 1000-dim ImageNet classification for all networks). A reduced number of parameters is important when deploying applications in resource-restricted environments. An alternative for VGG models is to discard the FC layers (similarly to our *avgpool5* descriptor), reducing the number of parameters from 134M to 14.7M.

Cosine and Bray-Curtis measures presented consistently good results among all experiments, including descriptors extracted from convolutional layers (*pool5*, *res5*, and *dense5* for VGG-16, ResNet-50, and DenseNet-161, respectively), with a few exceptions where Canberra outperforms them by a few percentage points. Cosine and Bray-Curtis tend to be more agnostic

Table 1. Retrieval performance on DTD for different combinations of DCNN architecture, feature extraction layer, and similarity measure.

Network	Layer (# of features)	Measure	mAP@1(%)	mAP@5(%)
VGG-16	*fc7* (4096)	Cosine	64.82	69.86
		Manhattan	61.47	66.83
		Bray-Curtis	63.08	68.27
		Canberra	63.15	68.20
	fc6 (4096)	Cosine	65.85	70.77
		Manhattan	60.91	65.95
		Bray-Curtis	65.57	66.85
		Canberra	65.67	70.36
	***avgpool5* (512)**	Cosine	67.23	72.04
		Manhattan	62.51	67.81
		Bray-Curtis	**68.20**	**72.68**
		Canberra	61.54	66.63
	pool5 (25088)	Cosine	63.79	68.89
		Manhattan	35.08	40.37
		Bray-Curtis	65.51	70.37
		Canberra	30.40	34.73
ResNet-50	***avgpool* (2048)**	**Cosine**	**70.61**	**75.06**
		Manhattan	68.80	73.03
		Bray-Curtis	70.20	74.71
		Canberra	67.35	71.64
	res5 (100352)	Cosine	68.06	72.51
		Manhattan	58.00	63.62
		Bray-Curtis	68.72	73.03
		Canberra	64.48	70.39
DenseNet-161	***avgpool** ***(2208)**	Cosine	72.32	76.52
		Manhattan	68.61	72.63
		Bray-Curtis*	**73.09***	**76.98***
		Canberra	72.62	76.16
	dense5 (108192)	Cosine	68.23	71.93
		Manhattan	63.09	67.8
		Bray-Curtis	68.72	73.03
		Canberra	71.13	74.94

**Best overall result.*

to the feature vector norm, and rather evaluate different aspects, which may explain their better coupling with DCNN-based features. Cosine similarity explores the angle between feature vectors in a hyperplane, which is norm independent. According to Kokare et al. [29], one of the main characteristics of Bray-Curtis dissimilarity is to give more attention to shared components with similar activations, instead of the position in a Cartesian hyperplane as in Euclidean distance. An advantage of the Bray-Curtis over cosine is requiring less floating point operations. In general, DCNN-based feature vectors that present activation on the same components are more similar than feature vectors that have activations in different components.

Qian et al. [14] also evaluated the performance of a CBIR system using the DTD dataset, extracting local features from the last convolutional layer of a VGG-M network pre-trained on ImageNet. They achieved mAP@1 of 62% and mAP@5 of 52% in their best scenario, using a Gaussian Mixture Model to model the image distribution, and evaluating Wasserstein distance to measure its similarities. These values are 11.09 pp and 24.98 pp smaller for mAP@1 and mAP@5, respectively, than our best pipeline results.

The classification task is much more common in texture analysis literature when compared to image retrieval. The best reported classification accuracy on DTD is 73.8%, achieved by Song et al. [22] with a 65 536-dim locally transferred DCNN-based Fisher Vector (LFVCNN) descriptor. This is comparable to our 73.09% mAP@1 result, obtained with a much smaller 2 208-dim descriptor.

Figure 3 illustrates the results for our best pipeline configuration. The texture content from retrieved images is quite similar to that of their respective queries in all cases. Also, as evidenced by the results in the first row, the DCNN descriptor has the ability to find similar textures in different orientations.

Figure 3. *Example results for the best pipeline configuration: DenseNet-161+avgpool+Bray-Curtis. Left-most images are query examples, and images on the right represent their respective top-5 retrievals.*

It is worth noting that some retrieved images with similar visual content are put in a different category from the query. This is due to the way the DTD dataset is constructed, which involves the subjective task of categorizing textures by their visual content. This discrepancy can lead to a decrease in the mAP@K evaluation metric, but is an expected drawback of adapting a classification dataset to the image retrieval task.

In order to evaluate the impact of dimensionality reduction on retrieval performance, we applied PCA on our best configuration (DenseNet-161, *avgpool*, Bray-Curtis). We varied the final number of components from 512 to 8 (Table 2).

Table 2 indicates that dimensionality reduction has a negative impact on retrieval performance. However, using 128 components reduces the dimensionality by approximately 94%, which can significantly lower storage and time requirements. This could

Table 2. Analysis of PCA dimensionality reduction impact on retrieval performance on DTD for DenseNet-161 *avgpool* 2208-dim descriptor using Bray-Curtis measure. Values in parentheses indicate reduction in percentage points from the uncompressed 2208-dim descriptor score.

# of Components	mAP@1(%)	mAP@5(%)
2208 (original)	73.09	76.98
512	68.30 (-4.79)	73.11 (-3.87)
256	69.27 (-3.82)	74.01 (-2.97)
128	**69.79 (-3.30)**	**74.25 (-2.73)**
64	69.53 (-3.56)	73.78 (-3.20)
32	67.23 (-5.86)	71.94 (-5.04)
8	45.66 (-27.43)	53.94 (-23.04)

compensate for the 3.30 pp drop in mAP@1 and 2.73 pp drop in mAP@5. Additionally, the large extent of dimensionality reduction with slight performance decrease indicates a moderate level of redundancy in the DCNN-based descriptor.

Conclusion

We demonstrated that using DCNNs to extract global descriptors for texture images yields promising results in the context of CBIR applications.

We compared different network architectures, layers, and similarity measures to build a CBIR pipeline. Our best results were achieved with DenseNet-161 architecture by extracting a 2208-dim feature vector from its *avgpool* layer and using Bray-Curtis dissimilarity measure for comparing descriptors. There are several important results from our experiments. Firstly, ResNet and DenseNet features exhibited superior performance over VGG-16, although VGG features are more commonly used in literature. Secondly, we demonstrated that applying global average pooling for VGG-16 convolutional features can improve over FC layers features. Thirdly, the cosine and the Bray-Curtis measures provided consistent better results for feature vector comparison. Additionally, we applied dimensionality reduction with PCA and we were able to reduce the number of components in the descriptor from 2208 to 128 while maintaining the retrieval performance.

For future work, we plan to evaluate more network architectures, given the increase in models with publicly available pre-trained weights. Another option is to train from scratch or fine-tune a network for a specific dataset. Additionally, we plan to measure the processing time under the usage of feature dimension reduction and different similarity measures, as this is a relevant issue for systems running in resource-restricted environments.

References

[1] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advance in content-based image retrieval: A literature survey. 2017.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2017.

[6] Hervé Jégou and Ondřej Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 774–787. Springer, 2012.

[7] Josef Sivic and Andrew Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1470–1477. IEEE, 2003.

[8] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–317, 2008.

[9] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311. IEEE, 2010.

[10] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed Fisher vectors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391. IEEE, 2010.

[11] Ran Tao, Efstratios Gavves, Cees G.M. Snoek, and Arnold W.M. Smeulders. Locality in generic instance search from one example. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2091–2098. IEEE, 2014.

[12] Ali S. Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *Journal of Transactions on Media Technology and Applications (MTA)*, 4(3):251–258, 2016.

[13] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pages 157–166. ACM, 2014.

[14] Y Qian, E Vazquez, and B Sengupta. Differential geometric retrieval of deep features. In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, pages 539–544. IEEE, 2017.

[15] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 241–257. Springer, 2016.

[16] Eva Mohedano, Kevin McGuinness, Noel E O'Connor, Amaia Salvador, Ferran Marqués, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pages 327–331. ACM, 2016.

[17] Joe Yue Hei Ng, Fan Yang, and Larry S. Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 53–61. IEEE, 2015.

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Dar-

rell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pages 675–678. ACM, 2014.

[19] Hang Zhang Jia Xue Kristin Dana. Deep TEN: Texture encoding network. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2896–2905. IEEE, 2017.

[20] Albert Jimenez, Jose M. Alvarez, and Xavier Giro-i Nieto. Class-weighted convolutional features for visual instance search. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–12, 2017.

[21] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3828–3836. IEEE, 2015.

[22] Yang Song, Fan Zhang, Qing Li, Heng Huang, Lauren J O'Donnell, and Weidong Cai. Locally-transferred Fisher vectors for texture classification. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4922–4930. IEEE, 2017.

[23] Li Liu, Songyang Lao, Paul W. Fieguth, Yulan Guo, Xiaogang Wang, and Matti Pietikäinen. Median robust extended local binary pattern for texture classification. *Transactions on Image Processing*, 25(3):1368–1381, 2016.

[24] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From BoW to CNN: Two decades of texture representation for texture classification. *International Journal of Computer Vision (IJCV)*, 126:1–36, 2018.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS-W*, 2017.

[27] Vincent Andrearczyk and Paul F. Whelan. Using filter banks in convolutional neural networks for texture classification. *Journal of Pattern Recognition Letters*, 84:63–69, 2016.

[28] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 420–434. Springer-Verlag, 2001.

[29] Manesh Kokare, B.N. Chatterji, and P.K. Biswas. Comparison of similarity metrics for texture image retrieval. In *Processing of the Conference on Convergent Technologies for Asia-Pacific Region (TENCON)*, pages 571–575. IEEE, 2003.

[30] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613. IEEE, 2014.

Author Biography

Augusto Valente *is a researcher at Eldorado Research Institute in Brazil. He received his B.Sc. in Electrical Engineering from the University of Brasilia (UnB) and is currently pursuing an M.Sc. degree at the State University of Campinas (UNICAMP), both in Brazil. His research interests include deep learning, computer vision and medical image processing.*

Fábio Perez *is a researcher at Eldorado Research Institute in Brazil. He received his B.Sc. in Computer Engineering from the State University of Campinas (UNICAMP), Brazil, and is currently pursuing an M.Sc. degree at the same institution. His research interests include deep learning, computer vision and medical image processing.*

Guilherme Megeto *is a researcher at Eldorado Research Institute in Brazil. He has a degree in Applied Mathematics and M.Sc. in Agricultural Engineering from State University of Campinas (UNICAMP), Brazil. His current research interests include deep learning, machine learning, computer vision and digital image processing.*

Marcos Cascone *is a researcher at Eldorado Research Institute in Brazil. He received his licentiate degree in Mathematics from São Paulo State University (UNESP), his M.Sc. in Statistics from the São Carlos Federal University (UFSCar) and currently he is a Ph.D. candidate in Statistics from Campinas State University (UNICAMP), both in Brazil. His research interests include deep learning, computer vision, time series and statistical modeling.*

Otavio Gomes *is a researcher at Eldorado Research Institute in Brazil. He received his B.Sc. in Computer Science from the Pontifical Catholic University of Rio Grande do Sul (PUCRS) and is currently pursuing an M.Sc. degree at the State University of Campinas (UNICAMP), both in Brazil. His research interests include computer vision, deep learning and 3D reconstruction.*

Thomas S. Paula *is a researcher working on deep learning and deep reinforcement learning at HP Labs. He has a degree focused on Web Systems Development and a Specialization on Applied Software Engineering, both from Feevale University, in Brazil. He received his M.Sc. in Machine Learning from the Pontifical Catholic University of Rio Grande do Sul (PUCRS), also in Brazil. His current research interests include deep learning, deep reinforcement learning, computer vision, and natural language processing.*

Qian Lin *is a distinguished technologist working on computer vision and deep learning research in HP Labs. Dr. Lin joined the Hewlett-Packard Company in 1992. She received her BS from Xian Jiaotong University in China, her MSEE from Purdue University, and her Ph.D. in Electrical Engineering from Stanford University. Dr. Lin is inventor/co-inventor for 44 issued patents. She was awarded Fellowship by the Society of Imaging Science and Technology (IS&T) in 2012, and Outstanding Electrical Engineer by the School of Electrical and Computer Engineering of Purdue University in 2013.*