Simple ConvNet Based on Bag of MLP-Based Local Descriptors

Takumi Kobayashi(✉), Hidenori Ide, and Kenji Watanabe

National Institute of Advanced Industrial Science and Technology,
Umezono 1-1-1, Tsukuba, Ibaraki, Japan
{takumi.kobayashi,hidenori.ide,kenji.watanabe}@aist.go.jp

Abstract. Deep convolutional neural network (ConvNet) is applied to versatile image recognition tasks with great success, though demanding high computation cost. Toward efficient computation, we propose a simple ConvNet architecture based on local descriptors in the bag-of-features framework. The local descriptors are formulated in a simple form of MLP and thus are efficiently computed on various ROI in a flexible manner. The proposed method is effectively trained in an end-to-end manner by reformulating the MLP descriptor into the form of deep ConvNet stacking convolution layers *linearly*. Through projection-based visual word encoding, the local descriptors are aggregated and fed into a classifier for image recognition tasks, which enables us to compute the network forwarding pass by matrix-vector multiplication. In the experiments on image classification, the proposed method is analyzed thoroughly, exhibiting favorable generalization performance on various tasks.

1 Introduction

Hand-crafted local descriptors, such as SIFT [15], extracted from small image patches have played a key role on various computer vision tasks; image classification was enthusiastically addressed by utilizing the descriptors in the bag-of-features (BoF) framework [9,22]. In this decade, however, deep convolutional neural networks (ConvNets) [11,26] defeat them with promising performance, though the hand-crafted descriptor is practically useful due to the low computation cost [30]. While the ConvNet works on whole input image through deeply stacked convolution operations, it is internally dependent on local image feature extraction directed by the last convolution, *e.g.*, at so-called `conv5` layer.

The local descriptors embedded in the deep ConvNets can be exposed and then combined with the traditional encoding schemes, such as Fisher kernel and bag of visual words, for image retrieval [16] and texture recognition [5]. There are also methods to leverage the ConvNet more directly to extract convolutional descriptors from image patches mainly on the task of patch matching [18,25]. Those ConvNet based descriptors are built on stacked convolution operations with computational burden [25], thus demanding sophisticated devices such as GPUs, unlike the hand-crafted SIFT. On the other hand, the hand-crafted descriptors are combined with neural network classifier of MLP through the Fisher kernel encoding

T. Gedeon et al. (Eds.): ICONIP 2019, CCIS 1142, pp. 207–215, 2019.
https://doi.org/10.1007/978-3-030-36808-1_23

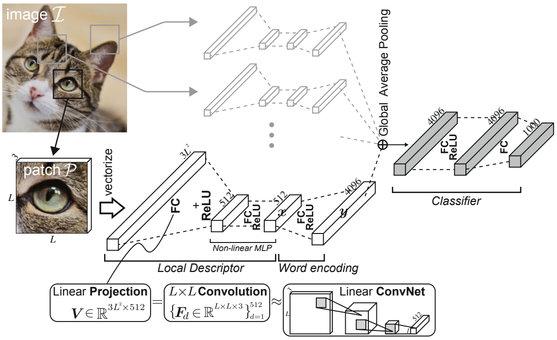

Fig. 1. Proposed network architecture based on MLP local descriptors.

Table 1. Baseline ConvNet architecture based on VGG16 [26].

	Block	Layers	Channel
Local descriptor ($L = 181$)	1	{3 × 3 Conv. + BatchNorm} ×2	64
		Down-sampling by 2-pixel stride	
	2	{3 × 3 Conv. + BatchNorm} ×2	128
		Down-sampling by 2-pixel stride	
	3	{3 × 3 Conv. + BatchNorm} ×3	256
		Down-sampling by 2-pixel stride	
	4	{3 × 3 Conv. + BatchNorm} ×3	512
		Down-sampling by 2-pixel stride	
	5	{3 × 3 Conv. + BatchNorm} ×3	512
		ReLU	512
	MLP	{1 × 1 Conv. + BatchNorm + ReLU}×0	512
	BoW	1 × 1 Conv. + BatchNorm + ReLU	4096
		Global Average Pooling	4096
	FC1	FC + BatchNorm + ReLU	4096
	FC2	FC	1000
		SoftMax	1000

in [19]. The method improves performance of the SVM classification approach [22], though being slightly inferior to AlexNet [11], which reveals the less discriminativity of the hand-crafted descriptor than the learned one.

In this work, we formulate a simple ConvNet toward efficient computation by explicitly considering the bag-of-features approach in the end-to-end framework. In contrast to the hybrid method [19] incorporating the hand-crafted descriptors with neural network classifier, we propose a simple form of local descriptor followed by visual word encoding, all of which are trained in an end-to-end manner as in the standard deep ConvNets. The simple architecture in descriptor design is based on MLP which comprises a linear projection and a non-linear function, *i.e.*, ReLU [17]; as a result, our model only requires matrix-vector multiplication efficiently computed by well established technique on various devices [8].

In the case that local patches are sampled at regular grids over an input image, the computation of our local descriptors, especially at the first layer of the MLP, can be regarded as convolution operation, thereby exhibiting similarity to the deep ConvNets. From the architectural viewpoint of ConvNets, however, the proposed model contrasts with the standard ConvNets as follows. The model based on the local descriptors contains only *one* convolution layer followed by several matrix-vector multiplication in MLP; the spatial convolution operates only on an input RGB image. Thus, from this viewpoint, our model is *less* convolutional compared to the *deep* ConvNets [2]. Such a simple computational procedure enables us to efficiently compute the forwarding pass of the model. In addition, it is possible to efficiently compute the local descriptors at regular grids by leveraging the convolution theorem [4] to perform the only one convolution via FFT. The other research line toward lightweight ConvNet is found in recent years [23,29]. While those works focus on slimming ConvNet still heavily relying on convolutional operation, we simplify the form of local descriptor through breaking dependence on the convolution to achieve computational efficiency as well as generalization performance.

On the other hand, the proposed model based on local descriptors in the BoF framework flexibly deals with any shape of ROI beyond simple regular grids unlike the standard ConvNets usually working on the regular lattice. The MLP-based feature extraction for local descriptors is also found in PointNet [20] to cope with point cloud data for 3D recognition. In this work, we employ an MLP model for computational efficiency and show favorable performance on image recognition tasks in spite of the simple formulation.

2 MLP-Based BoF Network

We build the neural network based on bag of local descriptors which are computed by applying multilayer perceptron (MLP) to local image patches, as shown in Fig. 1. Thus, computation for this network is simply composed of ReLU [17] and matrix-vector multiplication which is well-established operation on various devices [8]. While the similar MLP architecture is found in small image classification such as for MNIST [12], in this work, we leverage it to extract features from local patches in the bag-of-feature framework. Following [10], the descriptor $\boldsymbol{x} \in \mathbb{R}^{512}$ is encoded into word representation $\boldsymbol{y} \in \mathbb{R}^{4096}$ via linear projection by the word vectors $\{\boldsymbol{w}_i\}_{i=1}^{4096}$ with ReLU;

$$y_i = \max[\boldsymbol{w}_i^\top \boldsymbol{x} - \rho_i, 0] = \mathtt{ReLU}(\boldsymbol{w}_i^\top \boldsymbol{x} - \rho_i), \quad (1)$$

where ρ_i is a threshold for assigning the i-th word weight y_i to the descriptor $\boldsymbol{x}$ on the basis of inner-product similarity. The word representation aggregated across patches is then finally fed into the MLP classifier.

2.1 Training MLP Descriptor Through Linear ConvNet

The network (Fig. 1) can be trained end-to-end as in the deep ConvNets [11,26]. It, however, would be problematic to directly train the MLP descriptor which contains large projection matrix $\boldsymbol{V} \in \mathbb{R}^{3L^2 \times 512}$ in the first fully-connected layer; it depends on the patch size $L \times L$, say $L = 29$, which is larger than the standard convolution size, *e.g.*, 3×3. Thus, we reformulate the first fully-connected projection into a tractable form by means of *ConvNet*. It should be noted that our model is trained in a form of deep ConvNet but is deployed as the MLP-based form which is equivalent to the deep ConvNet.

In the descriptor MLP, the first fully-connected linear projection is viewed as *convolution* (without sliding) with the filters whose size corresponds to the patch size $L \times L \times 3$; this is just a transformation of the projection matrix $\boldsymbol{V}$ via unfolding. The moderately large $L \times L$ spatial filters are difficult to adequately learn due to the high degree of freedom (DoF). To mitigate it, we explicitly impose *decomposability into local convolutions* on the $L \times L$ convolution filters. This constraint is well validated by the Fractal structure, wavelet analysis and recent advances in deep ConvNet for image recognition. Thereby, the $L \times L$ convolution is approximated by stacking smaller convolutions, which results in the form of *linear* ConvNet (Fig. 1 & Table 1) without any non-linear functions, *e.g.*, ReLUs; a linear deep model is not *bad* even from the optimization viewpoint [7]. The linearly stacked convolution layers are compressed into a single convolution layer by enlarging the convolution filter as follows. Given two stacked convolutions whose filters are F of $l_F \times l_F$ and G of $l_G \times l_G$, we can describe the first convolution layer followed by down-sampling with factor s as

$$\tilde{I}(\boldsymbol{p}) = \sum_{\boldsymbol{\delta} \in \mathbb{Z}^2} F(\boldsymbol{\delta}) I(\boldsymbol{p} - \boldsymbol{\delta}), \;\; J(\boldsymbol{p}) = \tilde{I}(s\boldsymbol{p}), \tag{2}$$

where $I, \tilde{I}$ and J are input, intermediate and output feature maps, respectively, where the pixel position is denoted by $\boldsymbol{p}$. Then, the second convolution layer is

$$\begin{aligned}\tilde{J}(\boldsymbol{p}) &= \sum_{\boldsymbol{\epsilon} \in \mathbb{Z}^2} G(\boldsymbol{\epsilon}) J(\boldsymbol{p} - \boldsymbol{\epsilon}) = \sum_{\boldsymbol{\epsilon} \in \mathbb{Z}^2} G(\boldsymbol{\epsilon}) \tilde{I}(s\boldsymbol{p} - s\boldsymbol{\epsilon}) \\ &= \sum_{\boldsymbol{\epsilon} \in \mathbb{Z}^2} \hat{G}(\boldsymbol{\epsilon}) \tilde{I}(s\boldsymbol{p} - \boldsymbol{\epsilon}) = \sum_{\boldsymbol{\delta}, \boldsymbol{\epsilon} \in \mathbb{Z}^2} \hat{G}(\boldsymbol{\epsilon}) F(\boldsymbol{\delta}) I(s\boldsymbol{p} - \boldsymbol{\epsilon} - \boldsymbol{\delta}) \\ &= \sum_{\boldsymbol{\eta} \in \mathbb{Z}^2} \underbrace{\sum_{\boldsymbol{\delta} \in \mathbb{Z}^2} \hat{G}(\boldsymbol{\eta} - \boldsymbol{\delta}) F(\boldsymbol{\delta})}_{\text{Compressed filter } H(\boldsymbol{\eta})} I(s\boldsymbol{p} - \boldsymbol{\eta}),\end{aligned} \tag{3}$$

where $\tilde{J}$ is the output feature map, and we use the dilated filter of $\hat{G}(\boldsymbol{\epsilon}) = G(\frac{\boldsymbol{\epsilon}}{s})$ if $\frac{\boldsymbol{\epsilon}}{s} \in \mathbb{Z}^2$ otherwise 0, and transform the variable as $\boldsymbol{\eta} = \boldsymbol{\delta} + \boldsymbol{\epsilon}$. The size l_H of the compressed filter H is $l_H = s(l_G - 1) + l_F$. Thus, the patch size L, hyper-parameter of the descriptor, is naturally determined according to the architecture of the linear ConvNet.

This linear ConvNet is followed by the *non-linear* MLP to extract discriminative descriptors. The MLP is implemented as NiN module [14] of 1×1 convolution + ReLU layers, and thus in the case of regularly sampling patches on an input image during training, we implement our network (Fig. 1) by deep ConvNet (*e.g.*, Table 1) to effectively train the local descriptors and BoW model in an end-to-end approach. Once the network is trained, the linear ConvNet part is compressed by (3) into the fully-connected layer to form MLP-based descriptors. And, for densely computing descriptors on an image as in training, the descriptor can be efficiently extracted by applying the convolution theorem [4] via FFT.

3 Experimental Results

We evaluate various configurations of the MLP-based local descriptor in our network by training the corresponding ConvNets on a ImageNet dataset of 1000 object classes. All the models are implemented by using MatConvNet [27] following the good practice provided; the stochastic gradient descent is applied with the learning rate decreasing in a log-scale from 0.1 to 0.0001 over 40 epochs, the momentum of 0.9, the weight decay of 0.0005 and the mini-batch size of 64 samples. We measure the performance of top-5 error rate by single center cropping [11] on the ImageNet validation set.

Table 2. Performance analysis on various configuration of the local descriptor. The performance is evaluated by top-5 error rate (%) on ImageNet validation set. The baseline architecture in Table 1 is sequentially updated by the one denoted in bold font from (a) to (f).

(a) Convolutions per block

Architecture	Error (%)
Table 1 [$L=181$]	31.18
{3 × 3 Conv. + BN}×2 [$L=125$]	29.31
{3 × 3 Conv. + BN}×1 [$L=63$]	**29.17**

(b) Down-sampling

Method	Error (%)
striding [$L=63$]	**29.17**
avg.-pool [$L=78$]	30.79

(c) Convolution Filter size

Filter size	Error (%)
3 × 3 [$L=63$]	29.17
5 × 5 [$L=125$]	**27.59**
7 × 7 [$L=187$]	28.12

(d) Depth of Linear ConvNet

# of block	Error (%)
5 [$L=125$]	27.59
4 [$L=61$]	**24.71**
3 [$L=29$]	**24.76**
2 [$L=13$]	30.61

(e) Degree of non-linearity

Depth in MLP	4 block [$L=61$]	3 block [$L=29$]
0	24.71	24.76
1	20.29	19.80
2	18.55	**18.00**

(f) Training form of descriptor

Form	Error (%)
linear ConvNet [$L=29$]	**18.00**
29 × 29 Conv. [$L=29$]	22.24

3.1 Quantitative Ablation Study

We modify the baseline ConvNet (Table 1), according to the following analyses with keeping the descriptor dimensionality as 512.

Number of Convolution. The baseline model (Table 1) contains 13 layers of 3 × 3 convolution, 2 or 3 layers per block, across five blocks. Table 2a shows that the performance is improved by decreasing the number of 3 × 3 convolutions per block in contrast to the non-linear ConvNet containing ReLUs [26]; only one 3 × 3 convolution per block works well.

Local Pooling. In the linear ConvNet (Table 1), the feature maps are simply down-sized by 2-pixel striding, since 2 × 2 local average pooling degrades performance as shown in Table 2b. The local pooling is composed of 2 × 2 average filtering and 2-pixel striding, which unfavorably increases the convolution layers harming performance as implied in Table 2a.

Convolution Filter Size. On the other hand, by moderately enlarging the convolution filter size, we can improve performance as shown in Table 2c; the 5 × 5 convolution produces the best performance. Note that at each block *one* 5 × 5 convolution is equivalent to *two* stacked 3 × 3 convolutions (Table 2a), which

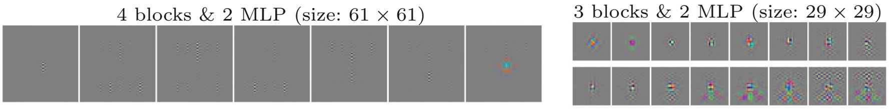
4 blocks & 2 MLP (size: 61 × 61)
3 blocks & 2 MLP (size: 29 × 29)

Fig. 2. The principal filters (columns of $\boldsymbol{U}_l$) by applying SVD to the learned filters.

conveys the insight that the larger-sized convolution in the shallower net is more effective than stacking smaller ones for a deep linear ConvNet.

Depth. Then, the number of blocks, *depth*, in the linear ConvNet stacking 5×5 convolutions is evaluated in Table 2d. The depth significantly affects the compressed filter size, *i.e.*, the patch size L. Compared to the larger patch descriptor, the moderate-sized ones produce the better performance; both the three ($L = 29$) and four ($L = 61$) blocks provide competitively good performance.

Non-linearity. The local descriptor is endowed with the non-linearity by the latter MLP part (Fig. 1 & Table 1) following the linear ConvNet part. Thus, the non-linearity is controlled by the depth of the MLP and Table 2e shows the performance improvement due to the higher non-linearity of the deeper MLP.

Training Form. As shown in Table 2f for training local descriptors, the form of linear ConvNet is superior to the naive MLP form, *i.e.*, one $L \times L$ convolution,

Based on the above analyses, we build the effective descriptor by stacking **three** 5×5 **convolution blocks** interlaced with the down-sampling of 2-pixel striding and **two-layer MLP**, which operates on a 29×29 patch with 4-pixel step for ImageNet classification. This configuration of the descriptor is closely related to the good practice [22] of the hand-crafted descriptor which extracts SIFT from 24×24 patches every 4 pixels on an image for image classification.

3.2 Qualitative Analysis

We qualitatively analyze the $L \times L$ spatial filter learned by the linear ConvNet.

For mining the principal characteristics in the spatial filters, we apply SVD to the (vectorized) filters $\boldsymbol{V} \in \mathbb{R}^{3L^2 \times 512}$ as $\boldsymbol{V} = \boldsymbol{U}_l \texttt{diag}(\boldsymbol{s}) \boldsymbol{U}_r^\top$; the filters are decomposed into 512 components, the columns of $\boldsymbol{U}_l \in \mathbb{R}^{3L^2 \times 512}$. As shown in Fig. 2, the deeper linear ConvNet of 4 blocks activates the filter weights only on a small spatial region due to the larger patch size, while the filter weights by the shallower one are diversely distributed. Thus, we can conclude that for constructing the effective linear convolutional features, it is necessary to design moderately deep (linear) ConvNet to provide a proper receptive field, followed by the highly non-linear MLP.

3.3 Generality

The proposed simple network exhibits superior performance (18.00%) to AlexNet [11] which produces 19.29% on the ImageNet dataset. We further show

Table 3. The performance comparison on various image classification tasks. The performance is measured by classification accuracy (%).

Type	Object		Scene		Other	
Dataset	VOC2007 [1]	Caltech256 [6]	SUN397 [28]	MIT67 [21]	FMD [24]	Event8 [13]
Ours	78.22	66.71	50.78	66.48	79.23	95.14
AlexNet	77.87	73.79	48.36	63.96	72.75	95.07
Hand-craft [9]	63.83	57.4	46.1	63.4	57.3	92.6

the generality of the descriptor-based simple network across various image recognition tasks. For that purpose, the model trained on the ImageNet dataset (Sect. 3.1) is transferred to the other datasets. For simplicity, the pre-trained network is applied to extract a 4096-dimensional image feature vector at FC1 (Table 1) which is followed by the linear SVM classifier. It is noteworthy that in our model, the descriptors are computed on 29 × 29 local patches every 4 pixels and then encoded into the word representation in a quite similar manner to the hand-crafted methods in the BoF framework [9,22]. For comparison, we employ the same procedure for the pre-trained AlexNet and also show the performances reported by the hand-crafted method [9] on the datasets of various image recognition tasks.

Table 3 shows the performance results on various tasks of image classification. As mentioned in [3], the AlexNet exhibits favorable transferability on object recognition tasks which are closely related to ImageNet classification. On the other hand, the proposed model produces superior performance even to the AlexNet on the other types of tasks while working competitively with the AlexNet on the object classification tasks. The network simply relying on the MLP-based local descriptors is endowed with such a better generalization performance. And, our method consistently outperforms the hand-crafted one [9] based on the SIFT-based descriptors, demonstrating that our descriptor trained end-to-end on ImageNet dataset is well discriminative with favorable generality.

4 Conclusion

We have proposed a simple network architecture for image recognition toward efficient computation. The proposed method is explicitly built upon the bag-of-features procedure which leverages local descriptors and visual word based representation to extract image features. While the descriptor is formulated by means of simple MLP for efficiency, it is effectively trained in an end-to-end manner through transforming the MLP into a form of ConvNet, by utilizing standard techniques/procedures tailored for deep ConvNets on ImageNet dataset. The proposed network mainly composed of simple MLP computation exhibits favorable performance not only on the ImageNet classification task but also on the other various image recognition tasks.

References

1. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). http://www.pascal-network.org/challenges/VOC/voc2007/index.html
2. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
3. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. PAMI **38**(9), 1790–1802 (2016)
4. Bracewell, R.N.: The Fourier Transform and Its Applications. McGraw-Hill, New York (1999)
5. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: CVPR, pp. 3828–3836 (2015)
6. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report 7694, Caltech (2007)
7. Kawaguchi, K.: Deep learning without poor local minima. In: NIPS, pp. 586–594 (2016)
8. Kestur, S., Davis, J.D., Chung, E.S.: Towards a universal FPGA matrix-vector multiplication architecture. In: FCCM, pp. 9–16 (2012)
9. Kobayashi, T.: Dirichlet-based histogram feature transform for image classification. In: CVPR, pp. 3278–3285 (2014)
10. Kobayashi, T.: Analyzing filters toward efficient convnets. In: CVPR, pp. 5619–5628 (2018)
11. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Li, L.J., Fei-Fei, L.: What, where and who? Classifying events by scene and object recognition. In: ICCV (2007)
14. Lin, M., Chen, Q., Yan, S.: Network in network. In: ICLR (2014)
15. Lowe, D.G.: Distinctive image features from scale invariant features. IJCV **60**, 91–110 (2004)
16. Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marques, F., Giro-i-Nieto, X.: Bags of local convolutional features for scalable instance search. In: ICMR, pp. 327–331 (2016)
17. Nair, V., Hinton, G.: Rectified linear units improve restricted Boltzmann machines. In: ICML, pp. 807–814 (2010)
18. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronnin, F., Schmid, C.: Local convolutional features with unsupervised training for image retrieval. In: ICCV, pp. 91–99 (2015)
19. Perronnin, F., Larlus, D.: Fisher vectors meet neural networks: a hybrid classification architecture. In: CVPR, pp. 3743–3752 (2015)
20. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: CVPR, pp. 77–85 (2017)
21. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR, pp. 413–420 (2009)
22. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. IJCV **105**(3), 222–245 (2013)
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNETV2: inverted residuals and linear bottlenecks. In: CVPR, pp. 4510–4520 (2018)

24. Sharan, L., Rosenholtz, R., Adelson, E.: Material perception: what can you see in a brief glance? J. Vis. **9**(8), 784 (2009)
25. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: ICCV, pp. 118–126 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
27. Vedaldi, A., Lenc, K.: MatConvNet - convolutional neural networks for MATLAB. In: ACM MM (2015)
28. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: CVPR (2010)
29. Zhang, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: CVPR, pp. 6848–6856 (2018)
30. Zheng, L., Yang, Y., Tian, Q.: SIFT meets CNN: a decade survey of instance retrieval. PAMI **40**(5), 1224–1244 (2018)