Earth Science Informatics (2020) 13:1437–1443
https://doi.org/10.1007/s12145-020-00484-3

METHODOLOGY ARTICLE

Exploiting low dimensional features from the MobileNets for remote sensing image retrieval

Dongyang Hou[1] · Zelang Miao[1] · Huaqiao Xing[2] · Hao Wu[3]

Received: 26 May 2020 / Accepted: 25 June 2020 / Published online: 29 June 2020

Abstract

Generally, traditional convolutional neural networks (CNN) models require a long training time and output high-dimensional features for content-based remote sensing image retrieval (CBRSIR). This paper aims to examine the retrieval performance of the MobileNets model and fine-tune it by changing the dimensions of the final fully connected layer to learn low dimensional representations for CBRSIR. Experimental results show that the MobileNets model achieves the best retrieval performance in term of retrieval accuracy and training speed, and the improvement of mean average precision is between 11.2% and 44.39% compared with the next best model ResNet152. Besides, 32-dimensional features of the fine-tuning MobileNet reach better retrieval performance than the original MobileNets and the principal component analysis method, and the maximum improvement of mean average precision is 11.56% and 9.8%, respectively. Overall, the MobileNets and the proposed fine-tuning models are simple, but they can indeed greatly improve retrieval performance compared with the commonly used CNN models.

Keywords Remote sensing image retrieval · MobileNets · Deep learning · High-dimensional features

Introduction

With the development of Earth observation technology, the number of high-resolution remote sensing (RS) images has grown rapidly(Bapu and Florinabel 2020; Shao et al. 2018). This has led to the challenge of efficiently retrieving objects or scenes of interest to users from the increased RS image database (Li and Ren 2017; Shao et al. 2020). Therefore, content-based remote sensing image retrieval (CBRSIR), which can rapidly acquire similar images from a large-scale dataset by using RS image features, has become research hotspots in the RS domain(Ge et al. 2018; Napoletano 2018).

Currently, a considerable literature has grown up around the theme of image feature extraction for CBRSIR. Initially, the mid/low level features are often directly extracted from RS images to represent their contents, such as HSV (hue, saturation, value) color space, bag of visual words, Gabor texture features and others (Du et al. 2016; Zhou et al. 2018; Zhou et al. 2015). Subsequently, various high-level deep learning features are becoming popular due to their high efficiency and effectiveness (Hou et al. 2019; Zhou et al. 2017). For example, Zhou et al. (2018) and Hou et al. (2019) employed various convolutional neural networks (CNN, i.e. AlexNet, VGG16, VGG19 and ResNet) to evaluate the performance of their CBRISR datasets, respectively. As described in the literature(Sudha and Aji 2019; Tong et al. 2019), scholars mainly use AlexNet, CaffeNet, VGG-M, VGG16, VGG19, GoogLeNet, ResNet, DenseNet and their variants or combination to carry out research on CBRSIR. Surprisingly, the effects of MobileNets networks, which is nearly as accurate as VGG16 in image classification while having less compute intensive(Howard et al. 2017), have not been closely examined in CBRSIR. In fact, experiments in literature(Qi et al. 2017) demonstrate that retrieval performance for natural images is improved by adding a hash layer to MobileNets compared to other hashing methods.

Communicated by: H. Babaie

Communicated by: H. Babaie

✉ Huaqiao Xing
xinghuaqiao@126.com

✉ Hao Wu
wuhao@ngcc.cn

[1] School of Geosciences and Info-Physics, Central South University, Changsha 410083, People's Republic of China

[2] School of Surveying and Geo-informatics, Shandong Jianzhu University, Jinan 250101, People's Republic of China

[3] National Geomatics Center of China, Beijing 100830, People's Republic of China

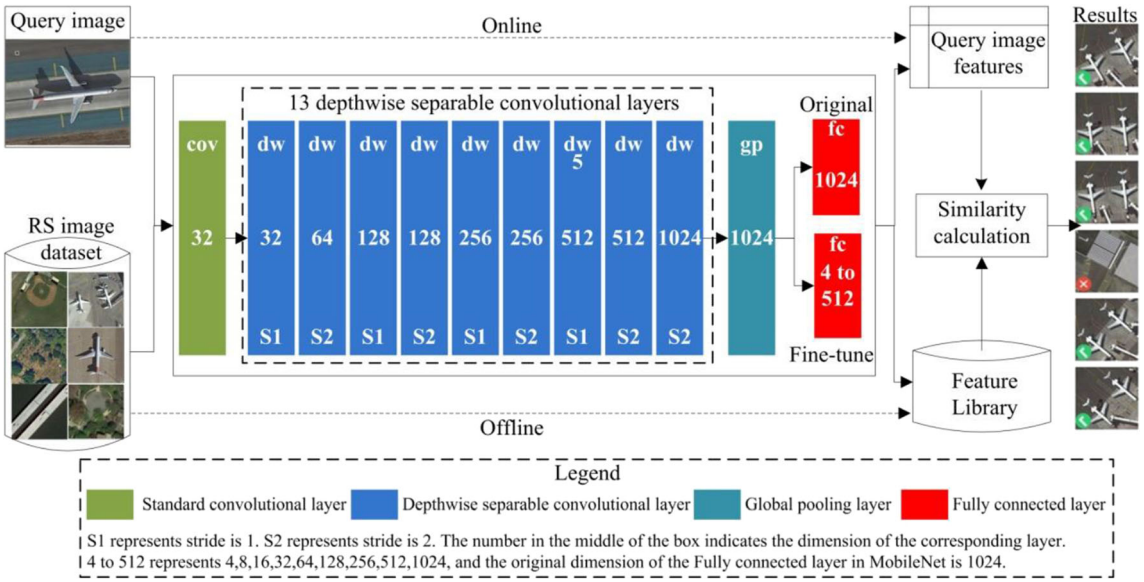

Fig. 1 Architecture of the original and fine-tuning MobileNets for CBRSIR

In general, the above high-level features directly extracted from deep learning methods are high dimensional with thousands of codes, which can lead to low retrieval efficiency, especially in a large image database(Ge et al. 2017; Tong et al. 2019). Therefore, several studies have attempted to compress high-level features as low dimensional features for better retrieval performance(Wang et al. 2020). For instance, Ge et al. (2017) used principal component analysis (PCA) method to compress CNN features to different dimensions and indicated that high-level features with 32 dimensions perform better. Tong et al. (2019) also demonstrated that the PCA method is effective for compressing CNN features and the optimized dimensions for CBRSIR are in the range of 8–32.

Unlike the above methods using the PCA compression, Xiao et al. (2017) treated the fully connected layers of CNN methods as ordinary neural networks and set 4096, 1024, 256, 64 dimensions of the second fully connected layer of Alexnet and VGG-16 to evaluate the retrieval performance. They concluded that the 64-dimensional features achieve the best retrieval results compared with other dimensional features and PCA-based features. Similarly, Cao et al. (2020) added a fully connected layer with a lower dimension in their proposed triplet network to condense the final features and also used PCA dimension reduction. Experimental results show that the PCA method has better performance than the fully connected-based method and the 32-dimensional features achieve the best retrieval results. Overall, there seems to be some evidence to indicate that the final fully connected layers can be treated as an ordinary neural network and directly modifying its dimension can achieve a similar dimensionality reduction effect as PCA methods(Cao et al. 2020; Hinton and Salakhutdinov 2006; Xiao et al. 2017). However, far too little attention has been paid to dimensionality reduction by modifying the dimension of the final fully connected layers in other deep learning methods.

Inspired by this and the efficient learning ability of the MobileNets, this paper investigates the retrieval performance of the MobileNets and exploits low dimensional features from the fine-tuning MobileNets for CBRSIR by changing the dimensions of the final fully connected layer. Our main contributions are as follows.

(1) We provide comprehensive comparisons between MobileNets and other commonly used deep learning

Table 1 Details of the six benchmark datasets used in this paper

Dataset	Class	Image number	Images per class	Sources	Size
NWPU	45	31,500	700	Google Earth imagery	256
AID	30	10,000	220–420	Google Earth imagery	600
PatternNet	38	30,400	800	Google Earth imagery	256
VArcGIS	38	59,071	1504–1904	ArcGIS World Imagery	256
VBing	38	58,944	1500–1880	Bing World Imagery	256
VGoogle	38	59,404	1502–1847	Google imagery map	256

methods on the six benchmark datasets, by giving a summary of retrieval performance and training time. Experimental results show that MobileNets achieves better retrieval performance than other CNN models while having shorter training time.

(2) We fine-tune the MobileNets to learn low dimensional representations by directly changing the dimensions of the final fully connection layer, and give the optimal dimensions of the fine-tuning model by experimental comparison. Experimental results indicate that 32-dimensional features achieve the best result, compared with the original MobileNets and PCA compression method.

The remainder of this paper is organized as follows. Section II outlines the methodological framework of the fine-tuning MobileNets, followed by extensive experiments and analysis in Section III. Section IV provides conclusions and future work.

Fine-tuning MobileNets networks for CBRSIR

MobileNets is a recent efficient CNN model, which is designed for various recognition tasks on mobile devices or under limited hardware conditions (Howard et al. 2017). It requires less computation than VGG16 model with only a small reduction in classification accuracy on the imagenet dataset (Howard et al. 2017). The reduction in classification accuracy may be the reason why no scholars have used the MobileNets in CBRSIR, whose main goal is to improve retrieval accuracy.

Figure 1 shows the architecture of the original and fine-tuning MobileNets for CBRSIR. Compared with other CNN models, it contains 13 depthwise separable convolutional layers and 13 pointwise convolutional layers, each of which is followed by each depthwise separable convolutional layer and is omitted in Fig. 1. Besides, each convolutional layers is followed by a batchnorm and ReLU nonlinearity. In the original MobileNets, the final fully connected layer is 1024 dimensions. In this paper, the final fully connected layer of the MobileNets is treated as output layer of ordinary neural networks and is fine-tuned to 512, 256, 128, 64, 32, 16, 8 and 4 dimensions, respectively, for learning low dimensional features. To evaluate the retrieval performance of the fine-tuning MobileNets, the PCA method is also adopted to compress the high dimensional features from the original MobileNets.

Experiments and analysis

The experiments are implemented by using the Keras library with TensorFlow backend in Python language, and performed on the same desktop with Intel Core 3.70 GHz i7-8700K processor and 2 NVIDIA GeForce GTX1080Ti GPUs.

Datasets and experimental setup

Six benchmark datasets of NWPU (Cheng et al. 2017), AID(Xia et al. 2017), PatternNet(Zhou et al. 2018), VArcGIS, VBing and VGoogle(Hou et al. 2019) are selected as the experimental data to demonstrate the retrieval accuracy of the MobileNets. Table 1 reports the details of these public datasets. As shown in Table 1, there are both datasets with the same source and different classification systems, as well as datasets with different sources and the same classification systems in the six datasets. This diversity can promote the credibility of evaluation results.

In total, six kinds of current state-of-the-art CNN models, which have been widely used for RSIR, are selected as comparison standard. In detail, our selections include VGG16, VGG19, ResNet50, ResNet101, ResNet152 and

Table 2 The results of the seven deep learning models on the six datasets

	PatternNet		AID		NWPU		Vgoogle		VArcGIS		VBing	
	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP
VGG16_f1	0.6383	0.3070	0.8159	0.1469	0.8689	0.1108	0.6681	0.2641	0.6869	0.2444	0.6702	0.2592
VGG16_f2	0.5031	0.4407	0.7183	0.2184	0.7421	0.2040	0.5493	0.3816	0.5712	0.3571	0.5424	0.3860
VGG19_f1	0.6625	0.2992	0.8229	0.1409	0.8681	0.1092	0.7101	0.2445	0.7216	0.2311	0.7045	0.2430
VGG19_f2	0.4468	0.5034	0.7308	0.2081	0.7381	0.2047	0.4822	0.4534	0.4945	0.4386	0.4740	0.4579
ResNet50	0.2913	0.6542	0.6885	0.2042	0.7189	0.1988	0.4066	0.5004	0.4535	0.4497	0.4009	0.5074
ResNet101	0.1208	0.8548	0.5916	0.2985	0.5778	0.3360	**0.1972**	**0.7498**	0.2188	0.7186	0.2136	0.7237
ResNet152	**0.1148**	**0.8625**	**0.5724**	**0.3177**	**0.5565**	**0.3592**	0.2014	0.7461	**0.2012**	**0.7403**	**0.1893**	**0.7550**
DenseNet	0.1834	0.7795	0.5895	0.2969	0.5857	0.3222	0.2921	0.6323	0.3294	0.5881	0.2961	0.6261
MobileNets	**0.0208**	**0.9745**	**0.1708**	**0.7616**	**0.2359**	**0.7151**	**0.0569**	**0.9255**	**0.0638**	**0.9128**	**0.0664**	**0.9117**

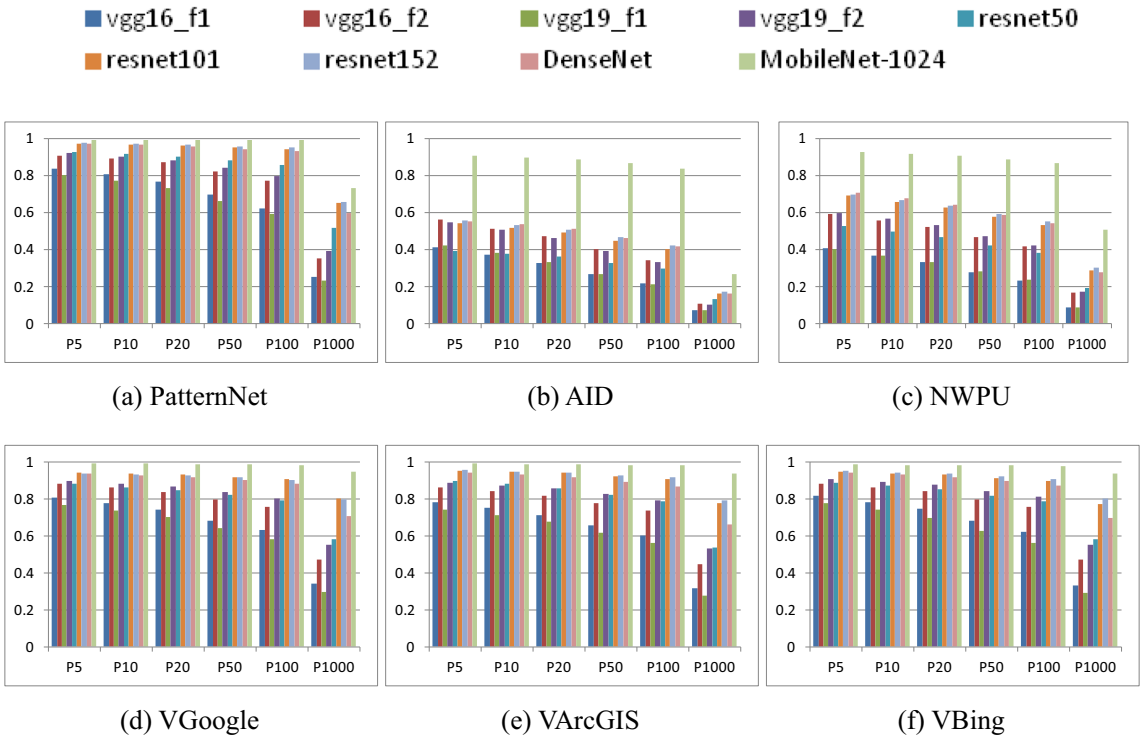

Fig. 2 Results of precisions at top 5,10,20,50,100 and 1000 on the six datasets

DenseNet201. In particular, the first and second fully-connected layers of VGG16 and VGG19 are both selected as features for comparisons, which are named as VGG16_f1, VGG16_f2, VGG19_f1 and VGG19_f2, respectively. For the ResNet and DenseNet201, the last global average pooling layer is selected as features.

In our experiments, the batch size is 32, the initial learning rate is 0.00001 and epoch number is set to 20 as described in literature (Tong et al. 2019). Besides, the most commonly used categorical cross entropy is selected as loss function to measure difference between actual output (probability) and the desired output (probability). 50 images from each class in the six datasets are randomly selected as query images and the remaining images are randomly split into a training set and a validation set, respectively. In particular, 50 images from each class are separated for validation set and the rest images are served as training set. Taking VGoogle dataset as an example, a total of 1900,1900 and 55,604 images are selected query images, validation set and training set, respectively.

Euclidean distance is used to measure similarity in our experiments. The nearer the distance between visual features of query image and other images is, the more similar these images are, and vice versa.

Average Normalized modified retrieval rank (ANMRR), mean average precision (mAP), precision at k (Pk, the percentage of the number of ground truth images within the top k position of the retrieval results), which are three kinds of standard retrieval measures, are adopted to evaluate the results(Cao et al. 2020). The k value is set as 5,10,20,50,100 and 1000 in this paper. Especially, lower values of the ANMRR indicate better retrieval performance, while for mAP and Pk, higher is better(Hou et al. 2019; Zhou et al. 2018).

Table 3 The training time of the seven deep learning models on the six datasets

	Training time (second)					
	PatternNet	AID	NWPU	VGoogle	VArcGIS	VBing
VGG16	4508	1567	4336	9246	9115	9163
VGG19	4686	1570	4579	9642	9569	9538
ResNet 50	4552	1576	4135	9330	9274	9445
ResNet101	6269	1831	6411	12,646	12,897	12,850
ResNet 152	8704	2460	8846	17,995	17,890	17,838
DenseNet	10,425	2246	7904	18,078	18,188	18,328
MobileNets	**2339**	**1533**	**2497**	**4806**	**5005**	**4770**

Investigating retrieval performance of the MobileNets

We perform several experiments to investigate retrieval performance of the MobileNets. Table 2 shows the performance of the seven deep learning models on the six datasets. The best performance of these models is achieved by the MobileNets on the six datasets. Except

Table 4 The results of different dimensions of the fine-tuning MobileNets

Dimensions	PatternNet		AID		NWPU		VGoogle		VArcGIS		VBing	
	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP	ANMRR	mAP
4	0.1670	0.8249	0.5204	0.4086	0.5096	0.4307	0.2643	0.7144	0.2329	0.7366	0.2448	0.7341
8	0.0347	0.9621	0.2982	0.6307	0.2731	0.6893	0.0373	0.9550	0.0543	0.9353	0.0439	0.9458
16	0.0108	0.9870	0.1677	0.7781	0.1590	0.8105	0.0202	0.9732	0.0237	0.9680	0.0250	0.9667
32	**0.0095**	**0.9882**	**0.1384**	**0.8094**	**0.1384**	**0.8307**	**0.0183**	**0.9746**	**0.0215**	**0.9700**	**0.0247**	**0.9659**
64	0.0121	0.9851	0.1403	0.8039	0.1611	0.8037	0.0221	0.9693	0.0246	0.9644	0.0286	0.9595
128	0.0122	0.9850	0.1425	0.7980	0.1822	0.7782	0.0298	0.9594	0.0317	0.9546	0.0349	0.9518
256	0.0137	0.9831	0.1687	0.7657	0.2022	0.7542	0.0387	0.9482	0.0392	0.9447	0.0481	0.9350
512	0.0176	0.9785	0.1669	0.7678	0.2227	0.7306	0.0487	0.9355	0.0517	0.9284	0.0582	0.9230
1024	0.0208	0.9745	0.1708	0.7616	0.2359	0.7151	0.0569	0.9255	0.0638	0.9128	0.0664	0.9117

for the MobileNets, the ResNet152 performs best. However, the mAP values of the MobileNets improve by 11.2% to 44.39% than the ResNet152, which indicates that the retrieval performance of the MobileNets is much higher than other CNN models.

Figure 2 shows the results of precisions at top 5,10,20,50,100 and 1000 on the six datasets. We can see that the MobileNets still performs much better than other models when only the top 5,10,20,50,100 and 1000 results are returned. The top 100 precisions of the MobileNets on the PatternNet, VGoogle, VArcGIS and VBing datasets all achieve between 97.71% and 99.07%, the other two datasets reach between 83.92% and 86.81%, while the top 100 precisions of other CNN models are between 21.28% and 95.02%.

To test the efficiency of the various models, we directly select training time under the same conditions as an evaluation indicator rather than floating-point operations (FLOPs). This is because that the actual training time of models with similar FLOPs can vary by at least one order of magnitude(Almeida et al. 2019). Table 3 represents the training time of the seven deep learning models on the six datasets. It can be seen that the MobileNets spends less training time than other models with a maximum difference of 4 times, especially for the larger-scale datasets of VGoogle, VArcGIS and VBing.

Overall, the above comprehensive comparisons further illustrate that the MobileNets achieves better retrieval performance than other deep learning models while being smaller training time.

Table 5 Different dimensions' precisions at top 5,10,20,50,100 and 1000 on VGoogle dataset

Dimensions	P5	P10	P20	P50	P100	P1000
4	0.8747	0.8751	0.8745	0.8723	0.8676	0.7646
8	0.9877	0.9879	0.9868	0.9857	0.9846	0.9686
16	0.9904	0.9905	0.9904	0.9901	0.9893	0.9788
32	**0.9937**	**0.9929**	**0.9925**	**0.9917**	**0.9907**	**0.9797**
64	0.9919	0.9915	0.9913	0.9905	0.9896	0.9778
128	0.9912	0.9903	0.9898	0.9889	0.9877	0.9716
256	0.9918	0.9910	0.9899	0.9886	0.9866	0.9639
512	0.9914	0.9904	0.9894	0.9876	0.9856	0.9564
1024	0.9919	0.9911	0.9890	0.9862	0.9837	0.9496

Exploiting low dimensional features from the fine-tuning MobileNets

To exploit low dimensional representations from the fine-tuning MobileNets, we conduct several experiments with different dimensions. Table 4 shows the results of different dimensions of the fine-tuning MobileNets. It can be seen that the best low dimensions of the fine-tuning MobileNets are 32. Specifically, the maximum improvement of the mAP value is 11.56% compared with the mAP of the original MobileNets. Besides, the result of 16, 64 and 128 dimensions are very close to the results of 32 dimensions.

To prove that the precision of the top retrieval results was not sacrificed in the fine-tuning MobileNets, we take VGoogle

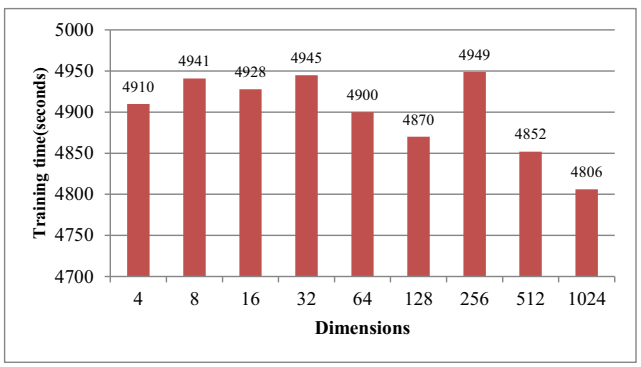

Fig. 3 Training time of different dimensions of the fine-tuning MobileNets on VGoogle dataset

Table 6 The results of 32 dimensions of the fine-tuning MobileNets and PCA-based method

Dataset	Methods	ANMRR	mAP	P5	P10	P20	P50	P100	P1000
PatternNet	Fine-tuning	0.0095	0.9882	0.9951	0.9943	0.9942	0.9934	0.9931	0.7429
	PCA	0.0151	0.9811	0.9934	0.9932	0.9929	0.9923	0.9913	0.7384
AID	Fine-tuning	0.1384	0.8094	0.8952	0.8866	0.8816	0.8716	0.8551	0.2671
	PCA	0.1388	0.8009	0.9045	0.9001	0.8927	0.8774	0.8559	0.2692
NWPU	Fine-tuning	0.1384	0.8307	0.9194	0.9156	0.9119	0.9042	0.8962	0.5717
	PCA	0.2199	0.7328	0.9167	0.9075	0.8990	0.8832	0.8661	0.5192
Vgoogle	Fine-tuning	0.0183	0.9746	0.9937	0.9929	0.9925	0.9917	0.9907	0.9797
	PCA	0.0445	0.9407	0.9907	0.9899	0.9893	0.9874	0.9854	0.9585
VArcGIS	Fine-tuning	0.0215	0.9700	0.9907	0.9898	0.9894	0.9877	0.9868	0.9757
	PCA	0.0493	0.9311	0.9905	0.9900	0.9884	0.9856	0.9828	0.9497
VBing	Fine-tuning	0.0247	0.9659	0.9876	0.9875	0.9864	0.9854	0.9839	0.9710
	PCA	0.0530	0.9284	0.9866	0.9858	0.9840	0.9819	0.9794	0.9469

dataset for example and give its different dimensions' results of precisions at top 5,10,20,50,100 and 1000 in Table 5. We can see that 32 dimensions of the fine-tuning MobileNets also achieve the best performance at top 5,10,20,50,100 and 1000 results, while it only takes around 2 min longer than the original MobileNets (as shown in Fig. 3).

Besides, we also adopt the PCA method to compress the high dimensional features from the original MobileNets into 32 dimensions for comparisons. Table 6 shows the results of 32 dimensions of the fine-tuning MobileNets and PCA-based method. It can be seen that the fine-tuning MobileNets offers a slightly better performance than PCA-based method and the maximum improvement of the mAP value is 9.8%.

Conclusions

In this paper, we examine the retrieval performance of the MobileNets model and fine-tune it by changing the dimensions of the final fully connected layer to learn low dimensional representations for CBRSIR. Experimental results indicate that the MobileNets outperforms other commonly used CNN models in term of retrieval accuracy and training speed. It also can be concluded that 32-dimensional features of the fine-tuning MobileNets achieves better retrieval performance compared with the original MobileNets and PCA compression method. Our future work will concentrate on exploiting low dimensional features from other MobileNets models and exploring their applications in multilabel remote sensing image retrieval.

Acknowledgments The authors would like to thank the PatternNet, NWPU and AID datasets for their open access. The authors also would like to thank the editors and the anonymous reviewers for their constructive comments and suggestions.

Funding information This work was supported in part by the National Natural Science Foundation of China under Grant 41,701,443 and Grant 41,801,308, and in part by the National Key Research and Development Program of China under Grant 2018YFB0505002.

Compliance with ethical standards

Conflict of interest No potential conflict of interest was reported by the authors.

References

Almeida M, Laskaridis S, Leontiadis I, Venieris SI, Lane ND (2019) EmBench: quantifying performance variations of deep neural networks across modern commodity devices. In: the 3rd international workshop on deep learning for Mobile systems and applications. Association for Computing Machinery, New York, pp 1–6. https://doi.org/10.1145/3325413.3329793

Bapu JJ, Florinabel DJ (2020) Automatic annotation of satellite images with multi class support vector machine. Earth Sci Inf. https://doi.org/10.1007/s12145-020-00471-8

Cao R, Zhang Q, Zhu J, Li Q, Li Q, Liu B, Qiu G (2020) Enhancing remote sensing image retrieval using a triplet deep metric learning network. Int J Remote Sens 41(2):740–751. https://doi.org/10.1080/2150704x.2019.1647368

Cheng G, Han J, Lu X (2017) Remote sensing image scene classification: benchmark and state of the art. Proc IEEE 105(10):1865–1883. https://doi.org/10.1109/jproc.2017.2675998

Du Z, Li X, Lu X (2016) Local structure learning in high resolution remote sensing image retrieval. Neurocomputing 207:813–822. https://doi.org/10.1016/j.neucom.2016.05.061

Ge Y, Jiang S, Xu Q, Jiang C, Ye F (2017) Exploiting representations from pre-trained convolutional neural networks for high-resolution remote sensing image retrieval. Multimed Tools Appl 77(13):17489–17515. https://doi.org/10.1007/s11042-017-5314-5

Ge Y, Tang Y, Jiang S, Leng L, Xu S, Ye F (2018) Region-based cascade pooling of convolutional features for HRRS image retrieval. Remote Sens Lett 9(10):1002–1010. https://doi.org/10.1080/2150704X.2018.1504334

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507. https://doi.org/10.1126/science.1127647

Hou D, Miao Z, Xing H, Wu H (2019) V-RSIR: an open access web-based image annotation tool for remote sensing image retrieval. IEEE Access 7:83852–83862. https://doi.org/10.1109/access.2019.2924933

Howard AG et al. (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861

Li P, Ren P (2017) Partial randomness hashing for large-scale remote sensing image retrieval. IEEE Geosci Remote Sens Lett 14(3):464–468. https://doi.org/10.1109/lgrs.2017.2651056

Napoletano P (2018) Visual descriptors for content-based retrieval of remote-sensing images. Int J Remote Sens 39(5):1343–1376. https://doi.org/10.1080/01431161.2017.1399472

Qi H, Liu W, Liu L (2017) An efficient deep learning hashing neural network for mobile visual search. In: 2017 IEEE global conference on signal and information processing (GlobalSIP), IEEE, Montreal, pp 701–704. https://doi.org/10.1109/GlobalSIP.2017.8309050

Shao Z, Yang K, Zhou W (2018) Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset. Remote Sens 10(6):964. https://doi.org/10.3390/rs10060964

Shao Z, Zhou W, Deng X, Zhang M, Cheng Q (2020) Multilabel remote sensing image retrieval based on fully convolutional network. IEEE J Sel Top Appl Earth Obs Remote Sens 13:318–328. https://doi.org/10.1109/JSTARS.2019.2961634

Sudha S, Aji S (2019) A review on recent advances in remote sensing image retrieval techniques. J Indian Soc Remote Sens 47:2129–2139. https://doi.org/10.1007/s12524-019-01049-8

Tong XY, Xia GS, Hu F, Zhong Y, Datcu M, Zhang L (2019) Exploiting deep features for remote sensing image retrieval: a systematic investigation. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2019.2948924

Wang Y, Ji S, Lu M, Zhang Y (2020) Attention boosted bilinear pooling for remote sensing image retrieval. Int J Remote Sens 41(7):2704–2724. https://doi.org/10.1080/01431161.2019.1697010

Xia G et al (2017) AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans Geosci Remote Sens 55(7):3965–3981. https://doi.org/10.1109/tgrs.2017.2685945

Xiao Z, Long Y, Li D, Wei C, Tang G, Liu J (2017) High-resolution remote sensing image retrieval based on CNNs from a dimensional perspective. Remote Sens 9(7):725. https://doi.org/10.3390/rs9070725

Zhou W, Shao Z, Diao C, Cheng Q (2015) High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder. Remote Sens Lett 6(10):775–783. https://doi.org/10.1080/2150704X.2015.1074756

Zhou W, Newsam S, Li C, Shao Z (2017) Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. Remote Sens 9(5):489. https://doi.org/10.3390/rs9050489

Zhou W, Newsam S, Li C, Shao Z (2018) PatternNet: a benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J Photogramm Remote Sens 145:197–209. https://doi.org/10.1016/j.isprsjprs.2018.01.004