Deep Convolutional Activations-Based Features for Ground-Based Cloud Classification

Cunzhao Shi, *Member, IEEE*, Chunheng Wang, Yu Wang, and Baihua Xiao

***Abstract*— Ground-based cloud classification is crucial for meteorological research and has received great concern in recent years. However, it is very challenging due to the extreme appearance variations under different atmospheric conditions. Although the convolutional neural networks have achieved remarkable performance in image classification, no one has evaluated their suitability for cloud classification. In this letter, we propose to use the deep convolutional activations-based features (DCAFs) for ground-based cloud classification. Considering the unique characteristic of cloud, we believe the local rich texture information might be more important than the global layout information and, thus, give a comprehensive evaluation of using both shallow convolutional layers-based features and DCAFs. Experimental results on two challenging public data sets demonstrate that although the realization of DCAF is quite straightforward without any use-dependent tricks, it outperforms conventional hand-crafted features considerably.**

***Index Terms*— Cloud classification, convolutional activations, convolutional neural network (CNN), fine-tune, max pooling, sum pooling.**

I. Introduction

CLOUD plays an important role in climate models, climate predictions, and meteorological services. However, the existing ground-based cloud classification is conducted by professionally trained observers [1], [2], which causes extensive human effort and might suffer from ambiguities due to the different standards of multiple observers. Therefore, automatic and efficient cloud classification is in great need.

A number of ground-based cloud image capturing devices have been developed to generate cloud images [3]–[7]. Based on the images acquired from these devices, researchers have proposed many effective hand-crafted features to deal with ground-based cloud classification. Buch and Sun [8] applied binary decision trees to classify the whole sky imager (WSI) images into five different sky conditions. Singh and Glennen [9] utilized co-occurrence and autocorrelation matrices for cloud classification. Fourier transformation was adopted by Calbó and Sabburg [10] to classify eight predefined sky conditions. Heinle *et al.* [11] proposed several statistical features for a fully automated classification of predefined seven sky conditions. Liu *et al.* [12] extracted some cloud structure features for infrared cloud images classification. Sun *et al.* [13] proposed to use local binary pattern (LBP) to classify infrared images. Liu *et al.* [14] further proposed the salient LBP for cloud classification. Recently, Dev *et al.* [15] proposed a modified texton-based classification approach that integrated both color and texture information for cloud classification and achieved better performance on the Singapore Whole-sky IMaging CATegories (SWIMCAT) database data set. However, most of the hand-crafted-based features have empirical parameters and could not deal with cloud images under different illumination conditions.

Recently, convolutional neural networks (CNNs) have shown remarkable performance in image classification [16]. CNNs have the ability to leverage large labeled data sets to learn increasingly complex transformations of the input to capture invariances. Importantly, CNNs pretrained on such large data sets have been shown to generate discriminative feature representations on many other domains [17], [18]. Recently, researchers have applied deep neural networks for object detection [19], [20] or land-use classification [21], [22] in very high-resolution remote sensing images. Domain transfer in CNNs is usually achieved by using the output of the fully connected (FC) layer of the network as features, which captures the overall spatial layout information. For cloud image representation, however, as the shape of the cloud is changeable, the deep convolutional features might be a better choice.

Although CNNs have achieved remarkable performance in image classification, no one has evaluated their suitability for cloud classification. As a special type of texture image, cloud image has its own characteristic. For cloud representation, the local rich texture information might be more important than the global layout information. Based on this assumption, in this letter, different from most CNNs-based image representation methods which usually adopt the FC layer and conv5 features, we pay more attention to the shallower convolutional layers from which local texture information could be extracted, and give a comprehensive evaluation of both the shallow and deep convolutional layers. Extensive experiments are conducted on two challenging public data sets. The results verify our assumption that the shallow layers could achieve similar or even better performance than the deeper layers. The results also demonstrate the suitability of deep

Manuscript received January 19, 2017; revised March 5, 2017; accepted March 9, 2017. Date of publication March 31, 2017; date of current version May 19, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61531019, Grant 61601462, and Grant 71621002. *(Corresponding authors: Cunzhao Shi; Chunheng Wang.)*

C. Shi, C. Wang, and B. Xiao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: cunzhao.shi@ia.ac.cn; chunheng.wang@ia.ac.cn).

Y. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Software, Shanxi University, Taiyuan 030006, China.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2681658

TABLE I
ARCHITECTURES OF TWO CNN MODELS USED IN THIS LETTER

Config.	conv1	conv2	conv3	conv4	conv5	fc6	fc7	fc8
Imagenet-vgg-f	$64 \times 11 \times 11$ max pool	$256 \times 5 \times 5$ max pool	$256 \times 3 \times 3$ max pool	$256 \times 3 \times 3$ max pool	$256 \times 3 \times 3$ max pool	4096 dropout	4096 dropout	1000 softmax
Config.	conv1-1 conv1-2	conv2-1 conv2-2	conv3-1 conv3-2 conv3-3	conv4-1 conv4-2 conv4-3	conv5-1 conv5-2 conv5-3	fc6	fc7	fc8
Imagenet-vgg-vd-16	$64 \times 3 \times 3$ $64 \times 3 \times 3$ max pool	$128 \times 3 \times 3$ $128 \times 3 \times 3$ max pool	$256 \times 3 \times 3$ $256 \times 3 \times 3$ $256 \times 3 \times 3$ max pool	$512 \times 3 \times 3$ $512 \times 3 \times 3$ $512 \times 3 \times 3$ max pool	$512 \times 3 \times 3$ $512 \times 3 \times 3$ $512 \times 3 \times 3$ max pool	4096 dropout	4096 dropout	1000 softmax

Fig. 1. Flowchart of the proposed method.

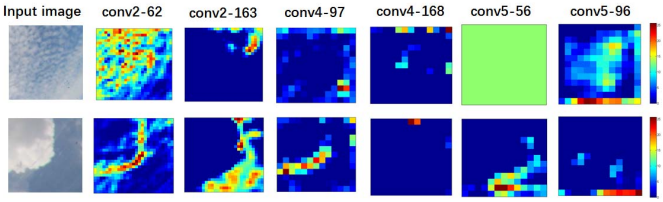

Fig. 2. Visualization of several feature maps from different convolutional layers of Imagenet-vgg-f.

convolutional activations-based features (DCAFs) for cloud classification, which outperforms conventional hand-crafted features considerably. Moreover, the realization of DCAF is quite straightforward without any use-dependent tricks and, thus, could be easily generalized to other data sets.

II. PROPOSED ALGORITHM

Fig. 1 shows the flowchart of the proposed method. The cloud images are directly fed into a CNN model; then, the features from different layers (either convolutional layers or FC ones) are extracted using different pooling strategies, and finally, a multilabel linear support vector machine (SVM) model is used to give the classification result. As we can see, the overall pipeline is quite straightforward without user-specific preprocessing stages and empirical parameters. In Sections II-A and B, we will first give some introduction of the deep CNN models we used and, then, detail the feature representation method for cloud image.

A. Deep Convolutional Neural Networks

CNNs have recently enjoyed a great success in large-scale image recognition [16] due to the large public image repositories and high-performance computing systems. Simonyan and Zisserman [23] proposed to steadily increase the depth of the network by adding more convolutional layers, resulting in very deep CNNs for large-scale image recognition. The deep CNNs have achieved state-of-the-art accuracy on imagenet large-scale visual recognition challenge (ILSVRC) classification and localization tasks, but are also applicable to other image recognition data sets. Therefore, in this letter, we use the deep CNN models to extract feature representation for cloud images.

The configurations of the two CNN models we used in this letter are outlined in Table I. For imagenet-vgg-vd-16, the input to the ConvNets is a fixed-size 224×224 red-green-blue (RGB) image. The only preprocessing is subtracting the mean RGB value, computed on the training set from each pixel. The image is passed through a stack of convolutional layers, where filters with a very small receptive field: 3×3 are used. The convolution stride is fixed to 1 pixel and the spatial padding of convolutional layer input is 1 pixel. All convolutional layers are equipped with the ReLU [16] nonlinearity. Spatial pooling is carried out by five max-pooling layers over a 2×2 pixel window with stride 2, following some of the convolutional layers. The convolutional layers are followed by three FC layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and, thus, contains 1000 channels (one for each class). The final layer is the soft-max layer. For cloud classification, the number of channels of the third FC layer depends on the categories of the cloud that need to be classified.

Imagenet-vgg-f is a simpler and faster version of imagenet-vgg-vd-16, which has less convolutional layers. It comprises five convolutional layers and three FC layers. The filters of the first convolutional layer have a larger receptive field: 11×11 than imagenet-vgg-vd-16 and the stride is also bigger: 4, which makes it faster to process the input image. The architecture of the three FC layer is similar to imagenet-vgg-vd-16.

The training of the original CNN model is carried out by optimizing the multinomial logistic regression objective using mini-batch gradient descent (based on backpropagation [24]) with momentum. The batch size is set to 256, momentum to 0.9. The training is regularized by weight decay (the L_2 penalty multiplier set to 0.0005) and dropout regularization for the first two FC layers (dropout ratio set to 0.5). Details of the training could be found in [16].

B. Deep Convolutional Activations-Based Feature for Cloud

As clouds have rich texture information, it is natural to describe cloud appearance using texture descriptors. Filter banks are powerful tools to extract texture features and have been widely used in texture analysis. These filter banks were designed to capture edges, spots, and bars at different scales and orientations [25]. CNN convolutional layers can be thought of as filter banks of increasing complexity with the depth [26]. The first layer extracts edge-like features and can be thought of as a filter bank approach, such as Gabor filters [27] or maximum response filters [25]. Intermediate

convolutional and pooling layers are analogous to filter banks extracting features of increasing complexity.

For a certain convolutional layer, suppose there are C filters to convolve with the input, resulting in C feature maps, each with the size of $H \times W$. Each filter is trained to get a high response for a certain type of pattern or texture. We visualize several feature maps from different convolutional layers of imagenet-vgg-f in Fig. 2. As we can see, different feature maps tend to have different activations for the same image, meaning that different filters are indeed trained to describe different patterns. Moreover, the activations of the same feature map for cloud images from different categories vary a lot, suggesting that the activations could be used as features to discriminate different categories.

Based on the aforementioned analysis, we use the convolutional activations-based features to represent cloud. Two pooling strategies are evaluated: sum pooling and max pooling. We employ sum-pooling scheme for each feature map, which is based on the aggregation of raw convolutional activation features. Let x_{ij}^k be the convolutional activations at position (i, j) from a certain feature map f_k, and the sum-pooling feature for map f_k is defined as

$$f_k = \sum_{i=1}^{h} \sum_{j=1}^{w} x_{ij}^k. \tag{1}$$

The overall sum-pooling DCAF (SDCAF) for a cloud image could be acquired by aggregating f_k from all the channels: $F = \{f_1, f_2, \ldots, f_c\}$. Then, we apply L_2-normalization on F to get the final representation.

Similarly, the max-pooling feature for map f_k is defined as

$$f_k = \max_{1 \le i \le H, 1 \le j \le W,} x_{ij}^k. \tag{2}$$

The complete max-pooling DCAF (MDCAF) is also acquired by aggregating f_k from all the channels: $F = \{f_1, f_2, \ldots, f_c\}$. F is then L_2-normalized.

For both SDCAF and MDCAF, the final feature dimension is related to the number of channels of a convolutional layer. As shown in Table I, for imagenet-vgg-f, the last three convolutional layers have 256 channels, whereas for imagenet-vgg-vd-16, the conv4 and conv5 layers have 512 channels.

For the FC layer-based feature, it could be regarded as a special case of convolutional layer, which has the kernel of size 1×1. The 4096-D features of FC layer are utilized for both models.

III. Experimental Results and Analysis

In this section, we first introduce the two databases for cloud classification. Then, we give the implementation details. Finally, we report the results of SDCAF and MDCAF from different layers of both models and compare our method with the traditional ones.

A. Database

1) SWIMCAT Database: SWIMCAT database contains images captured using wide angle high-resolution sky imaging system, a calibrated ground-based WSI designed by [28]. A total of 784 patches comprising five cloud categories are selected from images that were captured in Singapore over the period January 2013 to May 2014. The five categories include clear sky, patterned clouds, thick dark clouds, thick white clouds, and veil clouds. All image patches are of 125×125 pixels. Fig. 3(a) shows some examples.

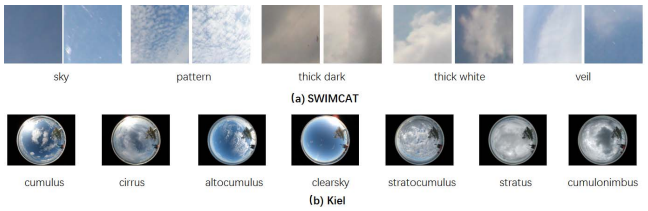

Fig. 3. Some samples from the two data sets.

2) Kiel Database: This database is provided by Kiel University in Germany [11]. The data set is divided into seven classes, including cumulus, cirrus, altocumulus, clear sky, stratus, stratocumulus, and cumulonimbus. It contains a total number of 1500 images with the size of 2272×1704. This data set has large illumination variations and intraclass variation. Fig. 3(b) shows samples from each category.

For both data sets, the samples are equally divided into several disjoint sets and each set is chosen as the test set, while the remaining ones constitute the training set. The final recognition accuracy is reported as the average accuracy of all the test sets. To evaluate the performance with increasing training samples, we report the results with 2–5 disjoint partitions.

B. Implementation Details

For imagenet-vgg-f, we evaluate four convolutional layers, namely, conv5, conv4, conv3, and conv2, respectively, and the FC layer, whereas for imagenet-vgg-vd-16, six convolutional layers, from conv4-1 to conv5-3, and the FC layer are evaluated. The images are normalized to 224×224 by bilinear interpolation and the mean RGB value is subtracted for each pixel. For all the various features, LIBLINEAR [29] is used for linear SVM training and classification with parameter settings "-s 2, -c 0.5." To fine-tune the model, we use the training set to retrain the model and evaluate it on the test set. Since the performance of the pretrained models on SWIMCAT is quite satisfactory, we only fine-tune the model on Kiel. The two-set partitions are evaluated. As imagenet-vgg-vd-16 achieves better performance on Kiel, we only fine-tune the model for imagenet-vgg-vd-16. The batch size is set to 64 and the training stops at 100 epoches. The learning rate for the first 40 epoches is set to 0.0001 and that for the remaining epoches is set to 0.00006. The training is regularized by weight decay (the L_2 penalty multiplier set to 0.0005).

C. Results and Discussion

1) SWIMCAT: The results of SDCAF and MDCAF using different layers of imagenet-vgg-f on SWIMCAT are listed in Table II and those using imagenet-vgg-vd-16 are shown in Table III. In each cell, the result of SDCAF is listed at the top and that of MDCAF at the bottom. F and C represent "Folds" and "Convolution," respectively. The results show that

TABLE II

RESULTS OF SDCAF AND MDCAF USING DIFFERENT LAYERS OF IMAGENET-vgg-f ON SWIMCAT AND KIEL

-	SWIMCAT					Kiel				
C / F	conv2	conv3	conv4	conv5	FC	conv2	conv3	conv4	conv5	FC
2	97.44 96.41	98.20 96.66	**98.72** 97.95	98.21 97.31	97.05	79.21 78.54	83.09 82.15	85.09 83.22	**85.56** 79.61	83.42
3	97.68 97.04	98.33 97.17	**98.46** 97.94	98.20 98.07	98.20	81.46 80.32	85.54 84.54	85.34 84.00	**86.08** 81.93	85.0
4	97.94 96.65	98.58 97.42	**98.84** 97.81	98.45 98.07	98.07	84.41 83.20	87.84 87.43	88.44 87.97	87.30 84.27	**88.58**
5	98.19 97.29	98.58 98.06	**98.84** 98.32	98.19 98.06	98.19	86.94 86.12	89.16 87.95	88.96 88.22	89.16 87.68	**89.90**

TABLE III

RESULTS OF SDCAF AND MDCAF USING DIFFERENT LAYERS OF IMAGENET-vgg-vd-16 ON SWIMCAT

C / F	conv4-1	conv4-2	conv4-3	conv5-1	conv5-2	conv5-3	FC
2	98.08 95.77	**98.46** 96.28	98.33 96.28	97.95 96.92	**98.46** 96.79	97.56 96.67	97.56
3	97.81 96.91	98.07 96.91	**98.20** 97.68	97.94 97.55	98.07 97.43	97.55 97.68	97.94
4	98.20 96.52	98.58 96.78	98.71 97.68	98.32 97.68	**98.97** 97.81	97.81 98.07	98.20
5	98.06 97.29	98.20 97.55	**98.32** 98.07	97.94 97.94	98.06 98.06	97.55 98.19	97.94

TABLE IV

RESULTS OF SDCAF AND MDCAF USING DIFFERENT LAYERS OF IMAGENET-vgg-vd-16 ON KIEL

C / F	conv4-1	conv4-2	conv4-3	conv5-1	conv5-2	conv5-3	FC
2	87.23 78.68	87.23 80.21	88.30 83.49	**88.84** 82.35	88.24 84.49	85.09 85.76	86.76
3	86.68 78.65	87.62 80.92	89.16 85.07	89.16 84.40	**89.36** 85.27	88.22 87.75	87.62
4	88.91 82.66	89.58 83.87	90.99 86.49	**91.13** 87.77	90.99 89.18	89.85 89.92	90.32
5	90.30 85.05	90.77 86.53	**91.58** 88.48	91.38 87.74	91.18 90.17	89.90 91.11	89.97

the FC layer fails to perform as well as the convolutional layers, which is reasonable, since the pretrained models are learned on a totally different data set and it captures more domain-specific overall spatial layout information, whereas the convolutional layers of the pretrained models achieve near-perfect performance with various data set partitions, reaching the classification accuracy of more than 98%, demonstrating the effectiveness of using the convolutional activations of a pretrained model to describe cloud images. Moreover, the relatively shallow layers (e.g., conv3 and conv2) could achieve comparable performance with deeper layers, suggesting that low-level orderless texture information is more important than high-level spatial layout to represent cloud images. Comparing Tables II and III, we find that the results of imagenet-vgg-f could even outperform those of imagenet-vgg-vd-16 slightly, further implying that for SWIMCAT, smaller models, such as imagenet-vgg-f, are enough to describe the textures.

2) Kiel: We list the results of SDCAF and MDCAF of imagenet-vgg-f and imagenet-vgg-vd-16 on Kiel data set in Tables II and IV, respectively. In each cell, the result of SDCAF is listed at the top and that of MDCAF at the bottom. The results show that the classification accuracy on Kiel is generally much lower than that on SWIMCAT, implying that

TABLE V

COMPARISON RESULTS OF DCAF WITH OTHER METHODS

Methods \ Folds	2	3	4	5	40/45
LBP(2,16)	85.26 68.32	81.60 68.54	83.51 80.58	85.03 83.23	93.47
Heinle feature [11]	90.26 59.49	91.89 59.37	92.91 59.34	93.42 62.22	93.09
Texton-based method [15]	-	-	-	-	95
DCAF	**98.72** **88.84**	**98.46** **89.36**	**98.97** **91.13**	**98.84** **91.58**	**99.56**

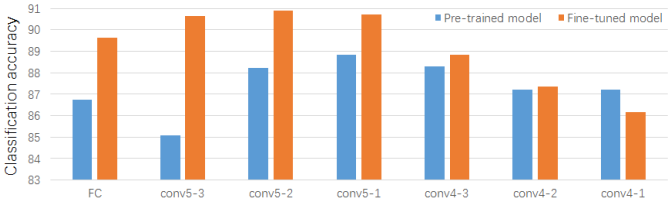

Fig. 4. Comparative results of pretrained model and fine-tuned one on Kiel.

Kiel exhibits more challenges than SWIMCAT as shown by the examples in Fig. 3. Moreover, different from SWIMCAT on which imagenet-vgg-f and imagenet-vgg-vd-16 achieve similar performance, the results of imagenet-vgg-vd-16 constantly outperform those of imagenet-vgg-f on Kiel with various layers and partitions, suggesting that deeper model, such as imagenet-vgg-vd-16, could extract more valuable patterns than shallow models on a more challenging data set. The results of various folds partitions also show that the performance improves with increasing training samples.

3) Fine-Tuning the Model: To demonstrate whether fine-tuning the model using cloud images could further improve the performance, we retrain the pretrained model imagenet-vgg-vd-16 with twofold partition. As SDCAF generally achieves better performance than MDCAF, we only report the results of SDCAF after fine-tuning. The results are shown in Fig. 4. As we can see, after fine-tuning, performance could be further improved and the improvement decreases when the layer gets shallower. For instance, the improvements of conv5-3, conv5-2, and conv5-1 are about 5%, 3%, and 2%, respectively, whereas for layers shallower than conv4-3, the improvement is very minor, suggesting that fine-tuning is more effective for the deeper layers which are nearer to the softmax layer, and thus, the parameters could be adjusted more effectively.

4) Comparative Results With Other Methods: We compare the proposed DCAF with the following methods: 1) the 12-D Heinle feature [11] which captures color, edge, and texture information of a sky/cloud image; 2) LBP, which is very effective for describing textures and has been used for cloud representation; and 3) the recently proposed texton-based approach [15], which shows the state-of-the-art performance on SWIMCAT. We test the results of LBP with $(P, R) = (8, 1), (16, 2)$, and $(24, 3)$ on the two data sets, where P is the number of sampling points on a circle of radius R and only list the results of $(P, R) = (16, 2)$ with which the best

performance is achieved. As the code for the texton-based approach [15] is not publicly available, we report the results with the same experimental setup as [15], which randomly selected 40 images per class for training and 45 ones for testing on SWIMCAT. We report the average accuracy of 50 random runs. The results along with those of DCAF are shown in Table V where "40/45" represents the setup in [15] on SWIMCAT. For each cell, the results of SWIMCAT are listed at the top and those of Kiel at the bottom.

The results show that the proposed DCAF outperforms Heinle feature and LBP significantly with various folds on both data sets. Moreover, with the same "40/45" setup, DCAF shows major superiority over Heinle feature and LBP, and even outperforms recently proposed texton-based approach [15] by more than 4%, demonstrating the advantages of DCAF-based features for cloud representation compared with traditional hand-crafted ones.

IV. Conclusion

In this letter, we propose to use the DCAF for ground-based cloud classification. Extensive experiments on two cloud image data sets demonstrate the suitability of using the activations of shallow convolutional layers of the pretrained deep CNN models for cloud representation. Moreover, the results show that the performance could be further improved after fine-tuning the pretrained models with cloud images. Comparative results show that DCAF outperforms traditional methods significantly, further demonstrating the major superiority of learned DCAF over hand-crafted features for cloud classification.

References

[1] J. Yang, W. Lu, Y. Ma, W. Yao, and Q. Li, "An automatic ground-based cloud detection method based on adaptive threshold," *J. Appl. Meteorol. Sci.*, vol. 20, no. 6, pp. 713–721, 2009.

[2] S. Liu, C. Wang, B. Xiao, Z. Zhang, and X. Cao, "Tensor ensemble of ground-based cloud sequences: Its modeling, classification, and synthesis," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1190–1194, Sep. 2013.

[3] J. E. Shields, "Daylight visible/nir whole-sky imagers for cloud and radiance monitoring in support of uv research programs," *Proc. SPIE*, vol. 5156, pp. 155–166, Nov. 2003.

[4] C. N. Long, D. W. Slater, and T. Tooman, "Total sky imager model 880 status and testing results," Pacific Northwest Nat. Lab. (U.S.), Richland, WA, USA, Tech. Rep. DOE/SC-ARM/TR-006, Nov. 2001.

[5] J. A. Shaw and B. Thurairajah, "Short-term arctic cloud statistics at NSA from the infrared cloud imager," in *Proc. 13th ARM Sci. Team Meeting*, Broomfield, CO, USA, Mar./Apr. 2003.

[6] A. Cazorla, F. J. Olmo, and L. Aladosarboledas, "Development of a sky imager for cloud cover assessment," *J. Opt. Soc. Amer.*, vol. 25, no. 1, pp. 29–39, 2008.

[7] X. J. Sun, T. C. Gao, D. L. Zhai, S. J. Zhao, and J. G. Lian, "Whole sky infrared cloud measuring system based on the uncooled infrared focal plane array," *Infr. Laser Eng.*, vol. 37, no. 5, pp. 761–764, 2008.

[8] K. A. Buch and C. H. Sun, "Cloud classification using whole-sky imager data," in *Proc. 9th Symp. Meteorol. Observ. Instrum.*, 1995, vol. 16, no. 3, pp. 353–358.

[9] M. Singh and M. Glennen, "Automated ground-based cloud recognition," *J. Pattern Anal. Appl.*, vol. 8, no. 3, pp. 258–271, 2005.

[10] J. Calbó and J. Sabburg, "Feature extraction from whole-sky ground-based images for cloud-type recognition," *J. Atmos. Ocean. Technol.*, vol. 25, no. 1, p. 3, 2008.

[11] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmos. Meas. Techn.*, vol. 3, no. 3, pp. 557–567, 2010.

[12] L. Liu, X. Sun, F. Chen, S. Zhao, and T. Gao, "Cloud classification based on structure features of infrared images," *J. Atmos. Ocean. Technol.*, vol. 28, no. 3, pp. 410–417, 2011.

[13] X. J. Sun, L. Liu, T. C. Gao, and S. J. Zhao, "Classification of whole sky infrared cloud image based on the LBP operator," *Trans. Atmos. Sci.*, vol. 32, no. 4, pp. 490–497, 2009.

[14] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, "Salient local binary pattern for ground-based cloud classification," *Acta Meteorol. Sinica*, vol. 27, no. 2, pp. 211–220, 2013.

[15] S. Dev, Y. H. Lee, and S. Winkler, "Categorization of cloud image patches using an improved texton-based approach," in *Proc. ICIP*, Sep. 2015, pp. 422–426.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *J. Adv. Neural Inf. Process. Syst.*, vol. 25, no. 2, pp. 1097–1105, 2012.

[17] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014.

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. CVPR*, 2014, pp. 1717–1724.

[19] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[20] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[21] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution Satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[22] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4238–4249, Aug. 2015.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2014.

[24] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[25] G. Caenen and L. V. Gool, "Maximum response filters for texture analysis," in *Proc. CVPRW*, 2004, p. 58.

[26] V. Andrearczyk and P. F. Whelan, "Using filter banks in convolutional neural networks for texture classification," *Pattern Recognit. Lett.*, vol. 84, pp. 63–69, Dec. 2016.

[27] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological*, vol. 61, no. 2, pp. 103–113, 1989.

[28] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "WAHRSIS: A low-cost high-resolution whole sky imager with near-infrared capabilities," in *Proc. IS, T/SPIE Infr. Imag. Syst.*, 2014, pp. 5450–5453.

[29] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.