Evaluation of Convnets for Large-Scale Scene Classification From High-Resolution Remote Sensing Images

Ratko Pilipović and Vladimir Risojević
Faculty of Electrical Engineering
University of Banja Luka
Bosnia and Herzegovina
{ratko.pilipovic, vlado}@etfbl.net

***Abstract*—Convolutional neural networks (convnets) have made possible a number of breakthroughs in image classification and other computer vision problems. However, in order to successfully apply convnets to a new task it should be trained on a large set of labeled samples. Acquisition of a large number of manually labeled remote sensing images requires highly trained analysts which makes it a very expensive task. This is the main reason why we still lack large training sets of remote sensing images. Nevertheless, convnets can still be applied to remote sensing image classification by virtue of using convnets pretrained on another large dataset and fine-tuned to the task at hand. In this paper we investigate the use of pretrained and fine-tuned convnets for both end-to-end classification and feature extraction from remote sensing images. We analyze the quality of the features extracted from various layers of the network from the standpoint of classification accuracy. Using a fine-tuned ResNet we obtain classification accuracy of over 94% on challenging AID dataset.**

***Keywords*—remote sensing image classification; convolutional neural networks; fine-tuning; feature extraction**

I. Introduction

Convolutional neural networks (convnets) have achieved state-of-the-art results and become the method of choice for various image classification tasks [1], [2]. The main obstacle in application of convnets to a new problem domain is the need for large labeled training sets. One of the areas where this problem can be observed is remote sensing image classification. Labeling of remote sensing images is expensive since it requires trained image analysts and, as a consequence, the available datasets of remote sensing images are considerably smaller than the datasets commonly used for training of convnets.

However, it has been shown that convnets trained on one image classification task can be used for feature extraction from images in a completely unrelated problem and still achieve excellent results [3]. This approach has also been successful in remote sensing image classification [4]–[6]. The most common technique for feature extraction using convnets has been to use the activations of the last fully connected layer before the softmax. However, it has been observed that activations of lower convolutional layers may be more appropriate as image features in scenarios where the distribution of images in the target task significantly differs from the distribution of images in the original task [7]. This is exactly the case when pretrained convnets are used for feature extraction in remote sensing image classification. Therefore, one of our goals in this paper is to analyze the quality of the features extracted from various layers of the network and identify the layer of a pretrained convnet which is the best choice for feature extraction in remote sensing image classification.

The performance of a pretrained convnet can be further improved through fine-tuning of the weights. During fine-tuning the topmost softmax layer is replaced with the new softmax layer having as many units as there are classes in the target task and the convnet is trained starting from the pretrained weights of the remaining layers. In this way, the weights are initialized to the values which ensure good performance on the original problem. As a result, the training converges to a better minimum of the cost function than when random weight initialization is used. It has been shown that fine tuning convnets pretrained on ImageNet yields good classification results on datasets of remote sensing images [5], [6]. Nevertheless, fine tuning is still very computationally intensive and it is unclear how a fine tuned network will cope with covariate shift caused by common variations in remote sensing images, such as using different sensors, changes in lighting and scale, as well as architectural differences in various parts of the world.

In this paper we evaluate two modern convnet architectures on the task of high resolution remote sensing image classification. We use convnets pretrained on ImageNet and aim to assess their applicability to remote sensing image classification. To this end we investigate fine-tuning of the weights on a dataset of remote sensing images. Next, we use

The research leading to these results has been co-funded by the European Commission under the H2020 Research Infrastructures contract no. 675121 (project VI-SEEM) and by the Ministry of Science and Technology of the Republic of Srpska under contract 19/6-020/961-37/15.

978-1-5090-3843-5/17/$31.00

convnets as feature extractors and train linear support vector machine (SVM) classifier using the extracted features. We extract several sets of features by pooling the activations from various convolutional layers and assess their suitability for remote sensing image classification.

The main contributions of this paper are: evaluation of modern convnet architectures on a challenging dataset of high-resolution remote sensing images, analysis of the features extracted from different layers of the network from the standpoint of classification accuracy, and analysis of the transferability of the features obtained using a fine-tuned convnet in the presence of variations common for remote sensing images.

The rest of the paper is organized in the following way. In Section II we review the previous work on using convnets for image classification in general, as well as remote sensing image classification. Then, in Section III we present the used convnets topologies and datasets. The classification results are presented and discussed in Section IV. The Section V is the conclusion.

II. Background and Related Work

Original concept of ConvNets dates back to Fukushima's "neocognitron" [8], the first artificial network that was invariant to image translations. Inspired by neocognitron, Lecun in [9] introduced LeNet, the first convnet architecture, and applied it to digit recognition. In spite of the promising results, convnets slowly gained popularity due to the lack of computational power and large labeled datasets needed for training. ImageNet dataset [10] with 1.2 million images divided into 1000 categories and availability of GPU implementations [1] are responsible for recent surge of popularity of convnets.

In order to improve performance in visual recognition, many convnet architectures have been proposed. In [11] it has been shown that significant improvement in classification accuracy can be achieved by increasing depth of a convnet. In [12] the topology of the network is crafted in such a way to increase both its depth and width. Furthermore, multiscale processing is used by introducing Inception modules which contain filters with different kernel sizes.

Although deeper neural networks show better results, they are hard to train and prone to overfitting, especially when small datasets are used for training. In order to overcome these shortcomings, in [2] the residual learning network (ResNet) was proposed. It contains shortcut connections bypassing convolutional layers which enables easier training of ResNets compared to plain deep convnets.

It was not only the rapid development of new convnet architectures that boosted the rise of convnets. In [3] it has been shown that a convnet trained on one image classification task can be used for extraction of features usable for various vision tasks, completely unrelated to the original one. In [7] the generalization ability of convnets was further investigated and fine-tuning of network feature filters to the task at hand was proposed.

Increasing popularity of convnets led to their use in remote sensing applications [13]. A number of approaches evaluated pretrained and fine-tuned convnets in high-resolution remote sensing image classification [4], [5], [14] and retrieval [15]. The question here is whether the features, extracted using a convnet pretrained on ImageNet dataset, can be used for classification of remote sensing images. In other words, are the features, extracted from pretrained or fine-tuned convnets, general enough to be used for classification of remote sensing images? In [6] a detailed evaluation of various approaches for utilizing convnets for remote sensing image classification is performed and SVM trained using features extracted from the fine-tuned convnet has shown the best performance.

The main obstacle to using convnets in remote sensing is the lack of large labeled remote sensing datasets, needed for training. To remedy this setback, in [16] two large datasets of remote sensing images, SAT4 and SAT6, are introduced. End-to-end training of modern convnet architectures on SAT4 and SAT6 is described in [17] and [18] and excellent results have been achieved. Altough SAT4 and SAT6 are large datasets and make end-to-end training of convnets possible, images from them represent very small patches of remote sensing images, and are divided into only 4 and 6 categories, respectively. It is still an open research question whether the convnets trained on SAT4 and SAT6 will be useful for classification of more complex high resolution remote sensing scenes.

The importance of benchmark datasets for further development in high resolution remote sensing scene classification is emphasized in [19]. A new challenging dataset with 10,000 images divided into 30 classes is proposed. We describe this dataset in Section III and use it for evaluation of two modern convnet architectures.

III. Material and Methods

In this Section we describe the used datasets and convnet architectures as well as the evaluation methodology.

A. Datasets

In this work we use two datasets of high resolution aerial images. The first one is AID dataset [19], a large scale remote sensing dataset which is constructed by collecting sample images from Google Earth. It contains 10,000 images divided into 30 aerial scene types: *airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residental, desert, farmland, forest, industrial, meadow, medium residental, square, stadium, storage tanks,* and *viaduct*. The number of images per class varies from 220 to 420. In order to increase intra-class variability the images are obtained from several countries, in different times and seasons, and under different imaging conditions. All images are RGB, 600×600 pixels with spatial resolutions ranging from about 8 meters to about 50 cm.

AID dataset addresses several shortcomings of the existing datasets. First, each class features a high diversity among sample images. This is mainly caused by imaging conditions,

e.g. the flying altitude and lighting condition which vary a lot. Secondly, AID dataset has small inter-class dissimilarity, in order to make it closer to real aerial image classification tasks. Images from different classes contain similar objects thus increasing demands for stronger generalization capability of a classification algorithm. Thirdly, AID dataset is the largest annotated aerial image dataset available to date, therefore it covers a broader range of aerial images and is a better benchmark for evaluation of image classification methods.

We also use UC Merced (UCM) dataset which consists of RGB images with the pixel resolution of one foot (30 cm). This dataset, first introduced in [20], was made from the United States Geological Survey (USGS) National Map images. It consist of 2100 images that are divided into 21 classes: *agricultural, airplane, baseball, diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks,* and *tennis courts*, with 100 images per class. UCM dataset is a long-standing benchmark in remote sensing scene classification. In this paper we use it to assess the transferability of features generated using a convnet fine-tuned on a different dataset of remote sensing images.

B. Convnet Architectures

Previous work [19] evaluated two traditional convnet architectures, namely CaffeNet and VGG_VD_16, as well as a more modern GoogLeNet, on AID dataset. In this work we evaluate two convnet architectures, which contain several new elements known to improve classification accuracy and convergence, as well as to reduce model size, on the task of large scale scene classification from high-resolution remote sensing images.

1) ResNet: Deep residual networks (ResNet) address the problem of training very deep convnets. More specifically, it has been observed [2] that, although deeper networks achieved better performances on image classification benchmarks, after a certain threshold adding more layers may result in higher training error. The main building blocks of ResNet are residual blocks with shortcut connections which bypass convolutional layers and make learning of the identity mapping easier as shown in Fig. 1. ResNets are constructed by stacking residual blocks and may be significantly deeper than the most successful architectures thus far. The experimental results show that these networks are easier to optimize and improve the accuracy on several computer vision benchmarks. Furthermore, in order to reduce the number of parameters, bottleneck building blocks have been proposed. In this paper, we use 50-layer ResNet architecture.

2) SqueezeNet: In [21] a convnet architecture that achieves AlexNet accuracy but with 50% less parameters and the model size smaller than 0.5 MB has been developed. The main building block of this architecture is *Fire* module which consists of two layers: *Squeeze* layer and *Expand* layer, as depicted in Fig. 2. Squeeze layer consists of 1×1 convolution filters, while Expand layer has a mix of 1×1 and 3×3 filters.

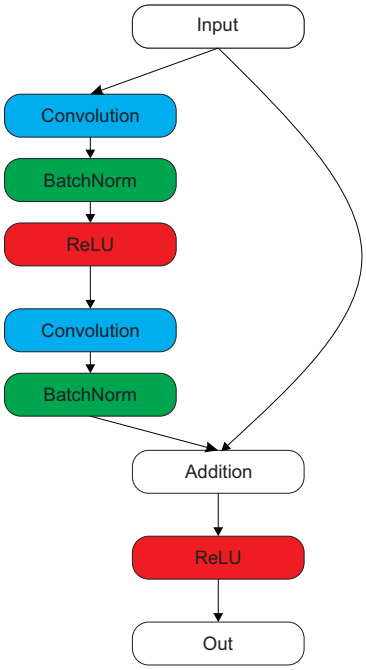

Fig. 1. Residual block.

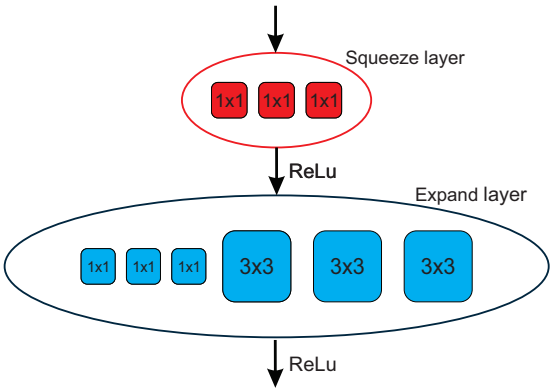

Fig. 2. The Fire module.

SqueezeNet architecture begins with a standalone convolution layer, followed by 8 Fire modules, and ending with a final convolution layer. After the first convolution layer and third Fire module, max pooling with stride 2 is applied. To prevent overfiting, dropout is applied after the last Fire module.

C. Evaluation Methodology

The main goal of this paper is to evaluate ability of convnets for scene classification in remote sensing images. To this end, we evaluate two approaches based on convnets pretrained on ImageNet. The first one is to fine-tune a convnet and use it for classification of remote sensing images, while the other is to use a convnet for feature extraction and then to use linear SVM for classification. In the first approach, we consider both the case when we freeze the weights of the hidden layers and only train the topmost softmax layer, and the case

when all the layers are fine-tuned. In the second approach, for feature extraction we use the pretrained network without as well as with fine-tuning on AID dataset. Feature vectors are constructed by spatial average pooling of feature maps at the output of a specific layer. Finally, we use the networks for feature extraction from UCM dataset and train SVM classifier. This last experiment is meant to investigate the ability of fine-tuned convnets to extract useful features from other datasets of remote sensing images subject to differences in resolution, sensors, imaging conditions, etc.

IV. Experimental Results

In the experiments presented in this section, we evaluate convnets for remote sensing image classification, as well as for extraction of features which are then used for training SVM classifier. For implementation of convnets we used Python and deep learning library Keras [22]. The experiments are performed on TITAN X GPU with 12 GB of RAM.

A. Classification Using Convnets

In the first set of experiments, we evaluate convnets for classification of images from AID dataset. We start with convnets pretrained on ImageNet dataset with 1000 object categories. Therefore, their topmost softmax layer has 1000 units. AID dataset has only 30 categories, and we replace the topmost softmax layer with 1000 units with another softmax layer with 30 units. Then, we consider two scenarios. First, we keep frozen the weights of all convolutional layers and train only the weights of the topmost softmax layer, and second, we fine-tune all the weights. In order to compare the results with [19], we randomly choose 50% of images for testing, 40% for training, and the remaining 10% for validation. We repeat the experiment for 10 random splits and report means and standard deviations of classification accuracy.

Although AID dataset is the largest manually labeled dataset of remote sensing images to date, the classification accuracies obtained using convnets trained from scratch are considerably lower than those obtained using pretrained convnets. Consequently, we did not include the experiments with convnets trained solely on AID dataset into this paper.

During the training we perform training data augmentation by randomly flipping images horizontally and vertically and by rotating them for a random angle between -180 and 180 degrees. We train the convnets using stochastic gradient descent with Nesterov momentum. The inital learning rate is 10^{-3} and we reduce it by a factor of 0.2 if the validation loss has not improved in the last 5 iterations. The obtained results are given in Table I. For comparison we also show the results from [19] which are obtained using the pretrained convnets for feature extraction and linear SVM for classification.

We can see that training only the softmax layer on top of SqueezeNet results in poor performance. However, fine-tuned SqueezeNet achieves the classification accuracy of nearly 89% which is comparable to the previous state-of-the-art in spite of the much lower number of parameters in SqueezeNet compared to the other convnets. An interesting result is obtained by training only the topmost softmax layer of ResNet. This strategy achieves classification accuracy of almost 90%, which is better than the previous best result. Even better result is obtained by fine-tuning ResNet and it outperforms the previous best result obtained using VGG_VD_16 for nearly 5%.

TABLE I
Classification accuracies (%) on AID dataset.

Model	Accuracy
SqueezeNet softmax	61.63 ± 2.51
SqueezeNet fine-tune	88.85 ± 0.83
ResNet softmax	90.62 ± 0.56
ResNet fine-tune	$\mathbf{94.23 \pm 0.34}$
CaffeNet [19]	89.53 ± 0.31
VGG_VD_16 [19]	89.64 ± 0.36
GoogLeNet [19]	86.39 ± 0.55

B. Feature Extraction Using Convnets

In the second set of experiments we use pretrained and fine-tuned convnets as feature extractors and subsequently classify the images using SVM. In order to get better insight into the quality of the features that can be extracted from various layers we adopted the following strategies. In SqueezeNet we extract features from the output of each Fire module, as well as from the outputs of the first and last convolutional layers, while in ResNet we extract features from the output of each group of residual blocks with the same number of filters. In all cases features are extracted using global spatial pooling of the activations of the specific layer. Therefore, the dimensionality of the feature vector is equal to the number of feature maps in that layer.

We randomly choose 50% of images for training the SVM classifiers and we test on the remaining 50%. The experiments are repeated 10 times with different random training/test splits and the classification accuracies are averaged.

The obtained results are given in Fig. 3 and 4. As expected, the classification accuracies obtained using the features extracted from fine-tuned convnets are higher than in the case of convnets trained on ImageNet only. Furthermore, in the case of fine-tuned convnets the accuracies keep increasing when features are extracted from higher layers. On the other hand, when convnet trained only on ImageNet is used, the classifier performs best for intermediate layers, and its performance drops as we move towards the higher layers. This behavior is a consequence of specialization of the filters in higher layers for detection of higher-level features specific to ImageNet. Since the images in ImageNet are very different from remote sensing images, those higher-level features are not discriminative enough for remote sensing image classification. When fine-tuned convnets are used, the discrimination power of the features increases as we move towards the higher layers of the convnet.

Finally, we examine whether the convnets fine-tuned on one dataset of remote sensing images will be useful for

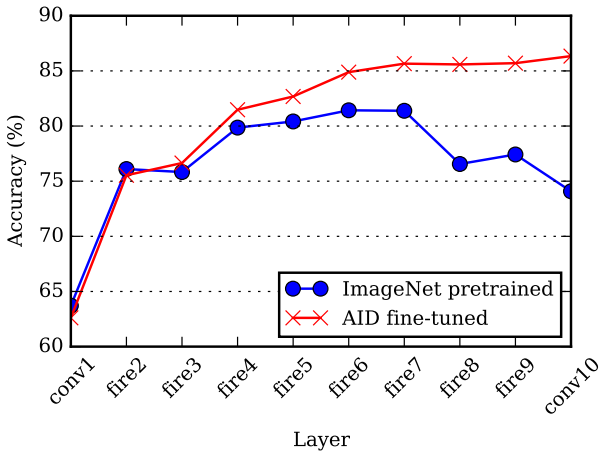

Fig. 3. The obtained classification accuracies on AID dataset using SqueezeNet for feature extraction and linear SVM for classification.

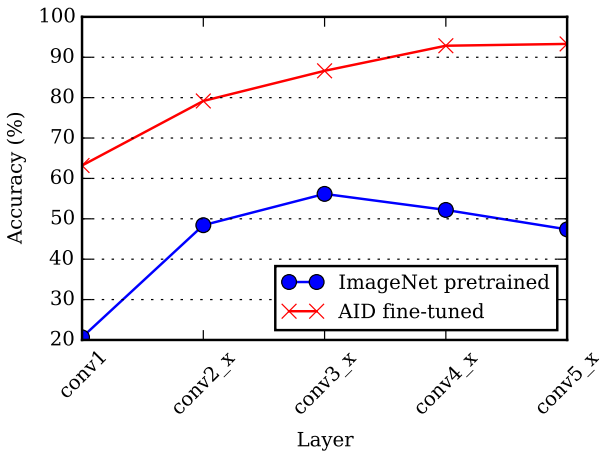

Fig. 4. The obtained classification accuracies on AID dataset using ResNet for feature extraction and linear SVM for classification.

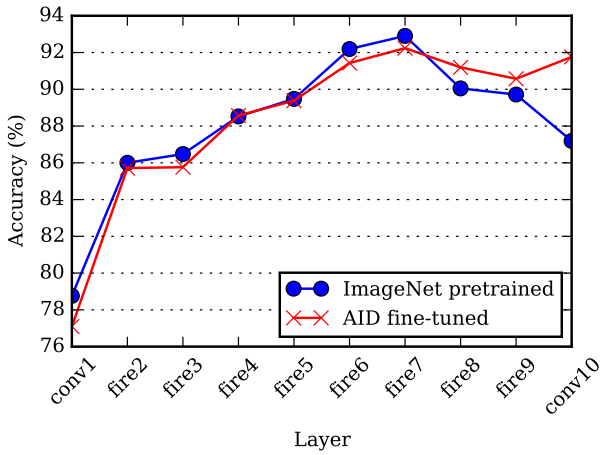

Fig. 5. The obtained classification accuracies on UCM dataset using SqueezeNet for feature extraction and linear SVM for classification.

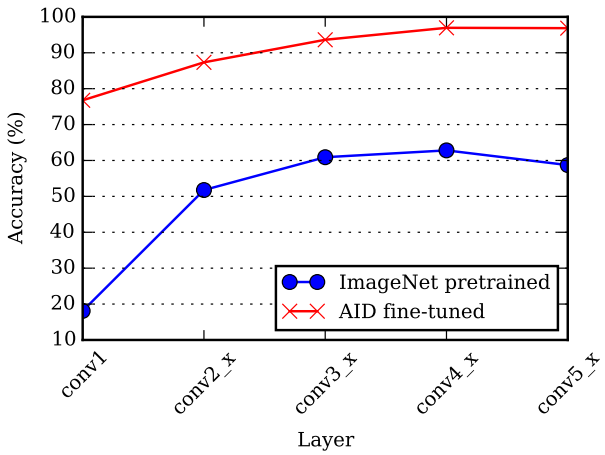

Fig. 6. The obtained classification accuracies on UCM dataset using ResNet for feature extraction and linear SVM for classification.

classification of images from the different dataset without further fine-tuning, i.e. are features extracted using convnets robust to covariate shift caused by variations in remote sensing images? To this end we extract features for images from the UCM dataset using convnets fine-tuned on AID dataset. For comparison purposes we also extract features using convnets trained on ImageNet only. As a classifier we again use linear SVM. The obtained results are shown in Fig. 5 and 6.

These results show that, when SqueezeNet is used the classification accuracies in both ImageNet pretrained and AID fine-tuned convnets are similar up to the output of the Fire7 module. This suggests that the features extracted using the modules before and including the Fire7 contain important discriminative cues for classification of images in UCM dataset. Having in mind that the images in ImageNet dataset are very different from remote sensing images it can be concluded that these are generic visual features and not specialized for a specific classification task. The classification accuracies obtained using features extracted from the modules after the Fire7 result in lower classification accuracies. This is more pronounced in the case of the convnet pretrained on ImageNet than in the case of the convnet fine-tuned on AID because the upper layers of the latter convnet still extract features which are more suitable for remote sensing image classification.

The classification accuracies obtained using features extracted from ResNet greatly differ in the cases when only the ImageNet pretrained network and the network fine-tuned on AID are used. It seems that residual blocks in any layer do not produce features which are universal enough to be transferred from object recognition to remote sensing image classification. In addition, the classification accuracy increases up to the fourth group of residual modules, and decreases afterwards. The reason is, again, the specialization of feature extractors for a task considerably different from remote sensing image classification. On the other hand, when the network is fine-tuned on AID dataset, the classification accuracies are higher for more than 30%. Furthermore, the classification accuracy increases after each group of residual blocks because the filter

TABLE II
COMPARISON OF CLASSIFICATION ACCURACIES ON UCM DATASET.

Model	Accuracy
SqueezeNet pretrained features	92.90 ± 0.65
SqueezeNet fine-tuned features	92.24 ± 1.12
ResNet fine-tuned features	96.95 ± 0.41
GoogLeNet fine-tuned features [6]	99.47 ± 0.50
GoogLeNet fine-tuned [6]	97.78 ± 0.97

TABLE III
ELAPSED TIMES FOR FINE-TUNING AND CLASSIFICATION PHASES.

Convnet	Epochs	Fine-tuning (epoch)	Classification (image)
SqueezeNet	174	40.81 s	0.8 ms
ResNet	74	52.77 s	10 ms

weights are fine-tuned to extract features which are relevant to remote sensing image classification.

In Table II we compare the best results on UCM with the results from the literature. It can be seen that the performance of descriptors extracted using the fine-tuned ResNet is very close to the state-of-the-art (GoogLeNet for descriptor extraction fine-tuned on UCM) in spite of the fact that we did not fine-tuned our networks on UCM.

Finally, in Table III we report timings for fine-tuning and classification phases for both convnet topologies, as well as the total number of epochs for fine-tuning. We can see that classification is very fast which opens up the possibility for various practical applications.

V. CONCLUSION

In this paper we evaluated modern convnets architectures on the task of remote sensing image classification. We experimentally showed that fine-tuned convnets can be effectively used both as classifiers as well as feature extractors. Using fine-tuned SqueezeNet we managed to obtain the classification accuracy of 88.85% which is close to the state-of-the-art but with significantly reduced number of parameters, while with fine-tuned ResNet we obtained classification accuracy of 94.23% which is the new best result on AID dataset.

The obtained results indicate that when ImageNet pretrained convnets without fine-tuning are used for feature extraction from remote sensing images it is better to use the activations of lower convolutional layers in order to achieve better generality of the extracted features. On the other hand, when fine-tuned convnets are used for feature extraction classification accuracy steadily improves when features are extracted from higher layers. Finally, we showed that convnets fine-tuned on AID dataset can be used for feature extraction from UCM and obtain near state-of-the-art results without further fine-tuning. This means that the network fine-tuned using remote sensing images is able to produce features useful for classification of remote sensing images from different datasets in spite of variations in sensors, resolution and imaging conditions.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[3] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519.

[4] O. A. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–51.

[5] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[6] K. Nogueira, O. A. Penatti, and J. A. d. Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *arXiv preprint arXiv:1602.01517*, 2016.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[8] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations*, 2015.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[13] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[14] I. Ševo and A. Avramović, "Convolutional neural network based automatic object detection on aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 740–744, 2016.

[15] P. Napoletano, "Visual descriptors for content-based retrieval of remote sensing images," *arXiv preprint arXiv:1602.00970*, 2016.

[16] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: a learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 37.

[17] M. Papadomanolaki, M. Vakalopoulou, S. Zagoruyko, and K. Karantzalos, "Benchmarking deep learning frameworks for the classification of very high resolution satellite multispectral data," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 83–88, 2016.

[18] Y. Zhong, F. Fei, Y. Liu, B. Zhao, H. Jiao, and L. Zhang, "Satcnn: satellite image dataset classification using agile convolutional neural networks," *Remote Sensing Letters*, vol. 8, no. 2, pp. 136–145, 2017.

[19] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang, "Aid: A benchmark dataset for performance evaluation of aerial scene classification," *arXiv preprint arXiv:1608.05167*, 2016.

[20] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.

[21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and less than 0.5 MB model size," *arXiv preprint arXiv:1602.07360*, 2016.

[22] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.