Analysis of Learned Features for Remote Sensing Image Classification

Vladimir Risojević, *Member, IEEE*

***Abstract*—Convolutional neural networks (convnets) have shown excellent results in various image classification tasks. Part of the success can be attributed to good image representations that are extracted using convolutional layers of the network. In this paper we consider convnets from the perspective of feature extraction for remote sensing image classification. We analyze the impact of convolutional feature extraction as well as the role of feature learning on the ability of features to discriminate between land cover classes. The quantitative analysis is based on measuring both classification accuracy and discriminative ability of features. For the latter we use Fisher discriminant analysis and show that features extracted using convolutional layers with random weights have significant discriminative ability and result in a reasonable baseline for remote sensing image classification, which suggests that convolutional feature extraction itself is an important ingredient of feature extraction in convnets. Using learned convnets for feature extraction further improves discriminative ability of features.**

***Keywords*—Feature learning, Fisher discriminant analysis, convolutional neural networks, remote sensing image classification.**

I. Introduction

Convolutional neural networks (convnets) have enabled significant breakthroughs in various image classification tasks and remote sensing image classification is not an exception to this trend [1], [2]. Traditionally, neural networks have been considered black boxes and trained end-to-end for a specific classification task. This has been one of the reasons for their success, because image representations and classifiers are learned in such a way that the most discriminative features are used for classification. However, at the same time it is also the main drawback of convnets because for successful end-to-end training we need large labeled datasets. Unfortunately, in remote sensing, data labeling is expensive and large labeled datasets are scarce.

Two results have made possible to circumvent the lack of training data and motivated the change in perspective on convnets. The first one is the observation that the outputs of a neural network with random weights can be used to train a classifier which will result in surprisingly good classification accuracy [3]–[6]. The second one is an interesting property of convnets that it is possible to obtain state-of-the-art classification accuracy on a given task by using a convnet trained on a completely unrelated task and only training the last layer for the task at hand [7]. This has also been successfully used in remote sensing image classification [8]–[11].

As a consequence of the above findings, we can regard a convnet as having two parts: (i) a feature extraction part, and (ii) a classifier part. This division is very loose and there is no strict rule which layers of the network constitute feature extraction and classifier parts. For the purpose of this paper we will consider the last layer of the network to be a classifier, and all the preceding layers to be a feature extractor.

In [12] the authors argue that features are the key ingredient to successful remote sensing image classification and propose a classifier which uses 22 hand-crafted features selected from a larger set of 150 features, based on distribution separability criterion. As a classifier a fully connected neural network is used. Trying to understand the reasons behind the obtained results, the authors show that the used features improve the separability of class-conditional distributions.

The main goal of this paper is to analyze convnets from the perspective of feature extraction for remote sensing image classification. To this end we evaluate the features obtained using convnets with one and two convolutional layers. We train and evaluate a simple softmax classifier using the obtained features and report classification accuracies on two datasets of satellite images. Our intention in this paper is not to obtain state-of-the-art results on the used datasets but rather to get better insight into the features extracted using a convnet. To this end we analyze the obtained features by assessing separability of the land cover classes in the feature space. As a measure of separability we use the ratio of between-class to within-class scatter in the feature subspace obtained using Fisher linear discriminant [13]. Similar criterion for comparing features was used in [14] where Fisher criterion was used to quantify the capability of a feature to discriminate between two textures, i.e. texture classes. In this paper we extend this approach to arbitrary number of classes. The main advantage of this approach is that the features are evaluated solely on the basis of their ability to discriminate between the classes, ruling out the influence of the specific classifier.

Since even neural networks with random weights can produce features that result in good classification accuracies, it follows that convolutional feature extraction per se plays an important role in obtaining a discriminative image representation. We examine this assumption by computing the Fisher criterion for features extracted using convolutional layers with random weights. Next, the impact

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).

The author is with the Faculty of Electrical Engineering, University of Banja Luka, 78000 Banja Luka, Republic of Srpska, Bosnia and Herzegovina (e-mail: vlado@etfbl.net)

Color versions of one or more of the figures in this paper are available online.

of learning is evaluated using convolutional layers from networks trained on the same or different classification problems.

The main contributions of this paper are:

- Evaluation of convnets for feature extraction in remote sensing image classification,
- Analysis of the discriminative ability of the obtained features regardless of the used classifier,
- Analysis of the impact of convolutional feature extraction on the ability of features to discriminate between land cover classes,
- Analysis of the role of learning in feature extraction.

The rest of the paper is organized in the following manner. In Section II basics of convnets are reviewed. The similarity of modern architectures for feature extraction and convnets is discussed in Section III. The datasets and methods used for the analysis are presented in Section IV. The experimental results are given in Section V. Section VI concludes the paper.

II. Convolutional Neural Networks

Convnets have arisen in the area of image classification and improved state-of-the-art in virtually every task they have been applied to. The main idea behind convnets is that images have pronounced local structure and joint distributions of spatially close pixels are more important than those of spatially distant ones. This fact has been confirmed by the discovery of local receptive fields of the neurons in mammalian visual cortex and leveraged in image classification through the use of Gabor filters, as well as local features such as SIFT, HOG, and LBP, for image description.

The output of a processing element in a fully connected layer depends on the values of all the pixels in an image. This means that the network has to learn the local structure of the image in order to use it effectively for image classification. Although it should be possible in general, it is very inefficient, so convnets explicitly encode local image structure by using a network topology in which a neuron in the hidden layer is connected only to the input neurons corresponding to spatially close pixels. The set of input neurons connected to a hidden neuron determines its receptive field, which is essentially a window in the input image. Each hidden neuron corresponds to a position of the window in the input image. An important ingredient of convnets is weight sharing, i.e. all hidden neurons use the same set of weights, since the same visual features appear in different positions in the input image. In this way the number of parameters is reduced compared to fully connected networks, which makes training easier and improves generalization. So, the hidden neurons act as feature detectors in various image locations. In essence, the input image is convolved with a filter defined by the set of weights of the neurons in a hidden layer and a feature map is obtained.

More formally, let $x \in \mathbb{R}^{n \times n}$ denotes an input image and $w \in \mathbb{R}^{k \times k}$ denotes a convolutional kernel then the feature map is obtained as

$$a = f\left(w * x + b\right), \tag{1}$$

where $b \in \mathbb{R}$ is a bias term and f is a nonlinear activation function. Many activation functions have been proposed, but the most used one at the moment is rectified linear function (ReLU)

$$a = f\left(z\right) = \max\left(0, z\right). \tag{2}$$

Image classification is performed on the basis of existence or absence of multiple features. Therefore, it is necessary to have multiple feature detectors. This is accomplished by using multiple filters each computing a feature maps of the form (1). Therefore, a hidden layer contains multiple feature maps and is referred to as convolutional layer.

Besides convolutional layers, convnets also contain pooling layers. They are commonly placed after convolutional layers. Neurons in pooling layers also have small receptive fields and its outputs are typically average (average-pooling or sum-pooling) or maximum values (max-pooling) of the activations of the neurons in the corresponding receptive field. A pooling layer discards the information about exact object positions and introduces a degree of invariance to small translations. Max-pooling layers are prevalent in modern convnets.

Convnets usually contain multiple convolutional and pooling layers followed by several fully connected layers. If a convnet is used for classification, the output layer is usually a softmax layer.

III. Convnets as feature extractors

Feature extraction for image classification today typically proceeds through a few stages as shown in Fig. 1. First, images are filtered using a filter bank. The filters can be either hand-crafted or learned using a variety of unsupervised and supervised approaches. After the filtering the obtained filter responses are encoded and finally pooled in order to obtain a discriminative image representation which is fed to a classifier.

For example, in the original bag-of-words (BoW) framework, computation of dense SIFT descriptors can be regarded as image filtering, because descriptors are computed for image patches sampled using a sliding window. Filter responses are then encoded against a codebook obtained by k-means clustering of the descriptors from the training set. Finally, the image representation is obtained by computing a histogram of codeword occurrences, which is essentially sum-pooling of one-hot codes. This framework has undergone many modifications which improved classification performance. For example, instead of k-means clustering and one-hot encoding, sparse coding has been introduced, max-pooling is used instead of sum-pooling, and unsupervisedly learned filters have shown performance comparable or better to hand-crafted SIFT descriptors.

From this standpoint, convolutional and pooling layers in a convnet can be regarded as a feature extractor and final fully connected layers as a classifier. More specifically, a convolutional layer is basically a filter bank and nonlinear activation function is equivalent to encoding of filter responses. Finally, pooling layers, as the name suggests, perform pooling of the obtained codes. The most striking

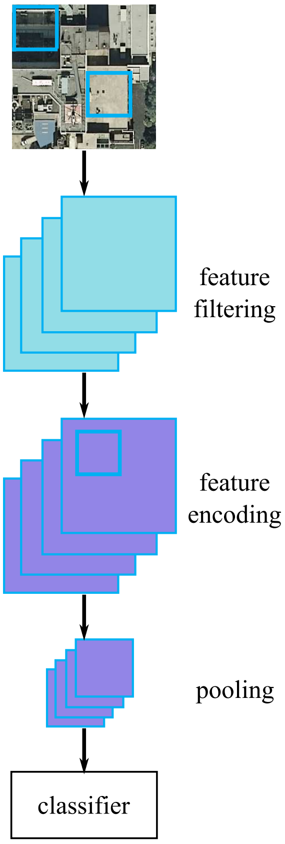

Fig. 1. Stages of convolutional feature extraction for image classification.

difference to the traditional feature extraction schemes is that filter weights, along with a classifier, are trained using a supervised learning algorithm. In this way image representations more adapted to the classification task at hand are obtained. As a result, convnets improved state-of-the-art results in virtually every image classification task they had been applied to.

In this standard setting, both the filters and the classifier parts of a convnet are jointly traiend. However, an interesting property of convnets has been observed which enforced the perspective that a convnet contains a feature extractor and a classifier. In various classification tasks it has been observed that convnets trained for completely unrelated tasks can yield state-of-the-art or better results with minimal fine tuning [7]. Fine tuning can, for example, mean learning the weights of the final softmax layer, while the weights of convolutional layers remain frozen. Furthermore, it has been also shown that the outputs of the last but one layer of a convnet can be effectively classified using SVM. In both the above scenarios, filter weights remain unchanged and only classifier is trained, which suggests that feature extraction is the task performed by convolutional layers.

As far as remote sensing imagery is concerned, the situation is even more interesting. Until recently, there were no labeled datasets of remote sensing images large enough for training of convnets. Nevertheless, in [8] it was shown that convnets pretrained on ImageNet can be used for classification of remote sensing images. Furthermore, in [15] it was shown that pretrained convnets can be used for feature extraction from remote sensing images and, in conjunction with SVM, yield competitive results, even without fine tuning. This result is remarkable having in mind large differences between the contents of ImageNet and remote sensing images.

IV. Material and Methods

In this paper we use SAT-4 and SAT-6 datasets [12]. The images in both datasets are sampled from the National Agriculture Imagery Program (NAIP) dataset, which consists of a total of 330,000 scenes spanning the whole of the Continental United States. These are the largest publicly available labeled datasets of remote sensing images. They contain images of size 28×28 pixels with spatial resolution of 1 meter and 4 spectral bands: red, green, blue and near infrared. SAT-4 contains 400,000 training and 100,000 test images manually classified into four land cover classes: barren land, trees, grassland, and a class that contains all other land cover classes different from the above three. SAT-6 contains 324,000 training and 81,000 test images manually classified into six land cover classes: barren land, trees, grassland, roads, buildings and water bodies.

For feature extraction we use convnets with one or two convolutional layers followed by pooling layers. Filters in the convolutional layers are 3×3 pixels, and the stride is equal to 1. We use max-pooling on non-overlapping regions of size 2×2 pixels. The filter weights are randomly initialized as described in [16]. We analyze the features obtained from both the convnets with random weights as well as from the convnets trained using backpropagation.

As a classifier, a single softmax layer is used. When random weights are used, only the classifier is trained, while the weights of the convolutional layers remain frozen. When the features are extracted using the trained convnet, we consider two scenarios. In the first scenario the convnet is trained on the same dataset that will be used for classification and the convolutional layers as well as the softmax layer are jointly trained using backpropagation. On the other hand, in the second scenario a different dataset is used for training, the weights of convolutional layers are then frozen, and the final classifier is trained, similarly to the case with random weights.

We train all convnets using stochastic gradient descent with Nesterov momentum. During the learning we monitor the validation error of the convnet and reduce the learning rate by half if the validation error did not drop for ten consecutive epochs. Following the reduction the learning rate is not reduced again for eight epochs. The learning is terminated if the validation error did not drop for 30 consecutive epochs or if the learning rate was reduced by a factor of more than 1000 in total.

The classifier is implemented using Theano and Lasagne and the experiments are performed on TITAN X GPU with 12 GB of RAM. In the experiments, we vary the number of filters and report the classification accuracies on the validation set.

In order to get better insight into the reasons behind the obtained classification accuracies, we further analyze the obtained features using Fisher criterion to evaluate the separability of classes in the feature space. The feature vectors for images from a single class form a cluster in the feature space. The better the separability of the clusters, the more the features are suitable for discrimination between the classes. The separability of the clusters depends both on their distance and compactness and can be assessed using Fisher discriminant analysis.

Given a set of d-dimensional samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, belonging to c classes, Fisher discriminant analysis finds their projections to $(c-1)$-dimensional space

$$\mathbf{y}_k = \mathbf{W}^T \mathbf{x}_k,\ k = 1, 2, \dots, n, \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d \times (c-1)}$ is the projection matrix obtained by maximizing the Fisher criterion which is the ratio of the between-class scatter and within-class scatter of the projected samples $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$. We define between-class scatter matrix

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i \left(\mathbf{m} - \mathbf{m}_i\right) \left(\mathbf{m} - \mathbf{m}_i\right)^T, \tag{4}$$

where n_i is the number of samples in the i-th class, $\mathbf{m}$ is the mean vector of all samples

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{y}_k, \tag{5}$$

and $\mathbf{m}_i$ is the mean vector of the set of feature vectors from the i-th class, $\mathcal{Y}_i$. Between-class scatter matrix is a measure of the distances between the clusters. We also define within-class scatter matrix

$$\mathbf{S}_W = \sum_{i=1}^{c} \sum_{y \in \mathcal{Y}_i} \left(\mathbf{y} - \mathbf{m}_i\right) \left(\mathbf{y} - \mathbf{m}_i\right)^T, \tag{6}$$

which is a measure of cluster compactness. Finally, total scatter matrix is defined as

$$\mathbf{S} = \mathbf{S}_B + \mathbf{S}_W. \tag{7}$$

The criterion function now has the following form [13]

$$J = \operatorname{tr}\left(\mathbf{S}^{-1} \mathbf{S}_B\right), \tag{8}$$

where $\operatorname{tr}(\cdot)$ is trace of a matrix. Since the trace is the sum of the eigenvalues of a matrix it measures the scattering volume in the direction of the eigenvectors. By maximizing (8) we tend to increase the between-class scatter and to decrease the within-class scatter. This is roughly equivalent to increasing the distance between the classes and, simultaneously, increasing their compactness. Therefore the larger the value of the Fisher criterion, the better the separability of the classes. The drawback of this approach is that the optimal solution is achieved only when the underlying distribution is Gaussian. However, in other cases the value of the Fisher criterion can still be regarded as an approximation of the class separability.

Fisher criterion was used in [14] for assessing the ability of Gabor-based features for discrimination between two textures, which is, essentially, a binary classification problem. In [12], distribution separability criterion (DSC) is used for measuring the discriminative power of features. DSC is computed as

$$D_S = \frac{\overline{\|\delta_{mean}\|}}{\overline{\delta_\sigma}}, \tag{9}$$

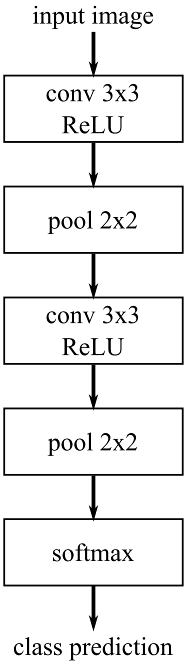

Fig. 2. An example of the convnet with two convolutional layers and without fully connected layers.

where $\overline{\|\delta_{mean}\|}$ is the mean of distances between means and $\overline{\delta_\sigma}$ is the mean of standard deviations of class conditional distributions. It can be seen that DSC is very similar to the Fisher criterion in the two-classes case. In this paper we consider an arbitrary number of classes. Therefore, we consider the Fisher criterion (8) to be a generalization of DSC.

V. Experimental Results

The goal of the experiments in this paper is to analyze how feature extraction using convolutional layers influences classification of remote sensing images into land cover classes. To this end we use convnets with one and two convolutional layers and no fully connected layers. When one convolutional layer is used, the number of units is chosen from the set $\{8, 16, 32, 64\}$. With two convolutional layers we evaluate two cases: (i) the number of units in the first layer is fixed to 8 and the number of units in the second layer is chosen from the set $\{8, 16, 32, 64\}$, and (ii) the number of units in the first layer is fixed to 32 and the number of units in the second layer is chosen from the set $\{32, 64\}$. Each convolutional layer is followed by a max-pooling layer, as described in Sec. IV. The topology of a convnet with two convolutional layers and without fully connected layers used in this paper is shown in Fig. 2.

When convnets are used for feature extraction it is important to assess both the importance of convolutional feature extraction as well as that of learning a set of filters. Therefore, we first performed experiments with random weights, and trained only the final softmax classifier. Then we trained the complete convnet and used its convolutional layers for feature extraction. Finally, we used the convolutional layers trained on SAT-4 for feature extraction in SAT-6 and vice versa. For both SAT-4 and SAT-6 we held 100,000 images from the training set for validation, and trained convnets on the rest.

TABLE I

Validation accuracies (%) on SAT-4 dataset obtained using features computed using convnets with one and two convolutional layers and varying number of convolutional units.

Architecture	Accuracy (random)	Accuracy (trained SAT-4)	Accuracy (trained SAT-6)
8c-mp	81.67	98.20	91.69
16c-mp	89.03	98.55	96.39
32c-mp	94.33	98.87	96.15
64c-mp	**96.82**	99.19	98.40
8c-mp-8c-mp	75.83	98.11	93.29
8c-mp-16c-mp	75.93	98.65	95.10
8c-mp-32c-mp	86.55	99.09	97.55
8c-mp-64c-mp	92.33	99.30	98.00
32c-mp-32c-mp	90.00	**99.52**	97.62
32c-mp-64c-mp	93.91	99.51	**98.51**

TABLE II

Validation accuracies (%) on SAT-6 dataset obtained using features computed using convnets with one and two convolutional layers and varying number of convolutional units.

Architecture	Accuracy (random)	Accuracy (trained SAT-6)	Accuracy (trained SAT-4)
8c-mp	96.36	98.50	98.64
16c-mp	97.20	98.65	99.04
32c-mp	97.70	98.89	99.15
64c-mp	**98.29**	99.21	99.30
8c-mp-8c-mp	94.25	97.62	98.95
8c-mp-16c-mp	96.01	98.39	99.01
8c-mp-32c-mp	97.08	98.82	99.29
8c-mp-64c-mp	97.56	99.28	99.35
32c-mp-32c-mp	97.14	99.23	**99.38**
32c-mp-64c-mp	97.79	**99.40**	**99.38**

TABLE III

Class separability on SAT-4 dataset computed using Fisher criterion vs. feature dimensionality for features extracted using convnets with one and two convolutional layers and varying number of convolutional units.

Architecture	J (random)	J (trained SAT-4)	J (trained SAT-6)
raw pixels	1.07		
8c-mp	1.28	1.89	1.52
16c-mp	1.67	2.11	2.10
32c-mp	1.99	2.42	2.28
64c-mp	2.28	2.63	2.52
8c-mp-8c-mp	1.07	2.27	1.46
8c-mp-16c-mp	1.46	2.43	2.12
8c-mp-32c-mp	1.59	2.48	2.24
8c-mp-64c-mp	1.88	2.58	2.45
32c-mp-32c-mp	1.84	2.56	2.26
32c-mp-64c-mp	2.06	2.58	2.47

TABLE IV

Class separability on SAT-6 dataset computed using Fisher criterion vs. feature dimensionality for features extracted using convnets with one and two convolutional layers and varying number of convolutional units.

Architecture	J (random)	J (trained SAT-6)	J (trained SAT-4)
raw pixels	2.22		
8c-mp	2.44	3.00	2.95
16c-mp	3.26	3.50	3.53
32c-mp	3.68	3.98	3.74
64c-mp	4.10	4.40	4.38
8c-mp-8c-mp	2.21	2.84	3.16
8c-mp-16c-mp	2.57	3.62	3.42
8c-mp-32c-mp	3.18	4.08	3.86
8c-mp-64c-mp	3.36	4.34	4.30
32c-mp-32c-mp	3.35	4.10	3.94
32c-mp-64c-mp	3.59	4.39	4.23

The validation accuracies obtained on SAT-4 dataset are shown in Table I. Several conclusions can be drawn. First, the features extracted using one convolutional layer with random weights make a reasonable baseline for classification of SAT-4 images. This is in line with the findings of [4] and [5]. Furthermore, the classification accuracy increases when more filters are used and the best result is obtained using 64 convolutional units followed by max-pooling. However, adding another convolutional layer with random weights does not necessarily improve the results, although increasing the number of filters in both layers again improves the classification accuracy.

Training the convnet for feature extraction improves the results considerably. In this case also, increasing the number of filters improves the classification accuracy. Most notably, adding another layer in this case also improves the results which suggests that the network learns a hierarchical representation of the data. We also tested the features extracted using the convolutional layers from the network trained on SAT-6 and the obtained results are close to those obtained using the convnet trained on SAT-4. Although the used datasets are very similar they contain different numbers of classes, which confirms that the extracted features are universal to some extent.

The validation accuracies obtained on SAT-6 dataset are given in Table II. Again, the similar conclusions can be drawn. In addition, in this case, the accuracies obtained with random filters are even closer to those obtained with learned filters. When filters learned on SAT-4 are considered, we can observe that the classification accuracies are even better than the accuracies obtained using filters learned on SAT-6, which further supports our assumption about the universality of learned features.

In order to get better insight into the features extracted using convnets we computed class separability measures (8) for all the cases discussed above. The results for SAT-4 are given in Table III, where the values of Fisher criterion are given for the tested convnets. When convnets with random weights are considered the reported values are obtained by averaging over five runs. We can see that, in all cases, the class separability computed using extracted features is better than when raw pixel values are used. Therefore, convolutional feature extraction improves the class separability thus making it is easier to find decision boundaries in the feature space. Adding more filters, even with random weights, improves class separability. On the other hand, adding more layers with random weights does not improve the class separability. Finally, training further improves separability of classes, which is consistent with the increase in accuracy. Unfortunately, it can be noticed that the class separabilities for random feature filters can be higher than the ones for the learned ones, although this is not the case with the corresponding classification accuracies. We believe that this is due to non-Gaussianity of the data, which makes Fisher criterion just an approximate measure of class separability. Similar conclusions can be made for the results for SAT-6 given in Table IV.

TABLE V
COMPARISON OF THE CLASSIFICATION ACCURACIES (%) ON THE TEST SET FOR SAT-4 AND SAT-6.

Model	SAT-4	SAT-6
DeepSat [12]	97.95	93.92
VGG [2]	99.98	99.98
64c-mp (random)	96.87	98.00
32c-mp-32c-mp (trained)	99.52	99.40
32c-mp-64c-mp (trained)	99.54	99.44

Although the evaluated convnet models were not designed to achieve state-of-the-art classification accuracy it is interesting to compare the obtained results to DeepSat [12] and state-of-the-art convnets [2]. To this end, in Table V the classification accuracies on the test set are given. We can see that in this case there is no need for unsupervised pretraining and even simple two-layered models examined in this paper can produce better results than the hand-crafted features.

VI. CONCLUSION

In this paper we examine the convnets from the perspective of feature extraction. On two datasets of satellite images we show that even convolutional layers with random weights can extract features which result in reasonable classification accuracies. By using convolutional layers from trained convnets classification accuracies increase. It is important to note that the convnet does not have to be trained for the same classification task and the results will still be close to the results obtained with the convnet trained on the same task. This suggests that the learned feature filters are universal to some extent.

We also analyzed the obtained features from the standpoint of class separability. To this end we used Fisher criterion and computed its value for features obtained using random as well as learned filters. From the results we conclude that convolutional feature extraction increases class separability compared to raw image pixels. Learning further improves the value of Fisher criterion. This essentially means that learned filters extract features in such a way that the distances between the samples from the same class will be smaller while, at the same time, the classes will be more distant in the feature space.

The obtained results show that there is no need for unsupervised pretraining if enough labeled images are available for training. Nevertheless, unsupervised pretraining might still be useful in cases with small number of labeled examples. Two interesting directions for future work arise from this research. The first one is the investigation of the universality of learned feature filters in the presence of different sources of variability in remote sensing images, and the other is concerned with avoiding non-Gaussianity of data and adapting Fisher discriminant analysis to work with features extracted using convnets.

REFERENCES

[1] 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.

[2] 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.

[3] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. Lecun, “What is the best multi-stage architecture for object recognition?” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2146–2153.

[4] A. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, “On random weights and unsupervised feature learning,” in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 1089–1096.

[5] A. Coates and A. Y. Ng, “The importance of encoding versus training with sparse coding and vector quantization,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 921–928.

[6] L. Liu and P. Fieguth, “Texture classification from random features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 574–586, 2012.

[7] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: An astounding baseline for recognition,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW ’14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 512–519. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2014.131

[8] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–51.

[9] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, “Land use classification in remote sensing images by convolutional neural networks,” *arXiv preprint arXiv:1508.00092*, 2015.

[10] 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.

[11] 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.

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

[13] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1990.

[14] S. E. Grigorescu, N. Petkov, and P. Kruizinga, “Comparison of texture features based on gabor filters,” *IEEE Transactions on Image processing*, vol. 11, no. 10, pp. 1160–1167, 2002.

[15] 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.

[16] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks.” in *Aistats*, vol. 9, 2010, pp. 249–256.