Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures

Article *in* The Journal of Supercomputing · November 2020

DOI: 10.1007/s11227-019-03106-y

Fatih Özyurt
Firat University

The Journal of Supercomputing
https://doi.org/10.1007/s11227-019-03106-y

Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures

Fatih Özyurt[1]

Abstract

Convolutional neural networks (CNNs) have recently emerged as a popular topic for machine learning in various academic and industrial fields. It is often an important problem to obtain a dataset with an appropriate size for CNN training. However, the lack of training data in the case of remote image research leads to poor performance due to the overfitting problem. In addition, the back-propagation algorithm used in CNN training is usually very slow and thus requires tuning different hyper-parameters. In order to overcome these drawbacks, a new approach fully based on machine learning algorithm to learn useful CNN features from Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet CNN architectures is proposed in the present study. This method performs a fast and accurate classification suitable for recognition systems. Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet pre-trained architectures were used as feature extractors. The proposed method obtains features from the last fully connected layers of each architecture and applies the ReliefF feature selection algorithm to obtain efficient features. Then, selected features are given to the support vector machine classifier with the CNN-learned features instead of the FC layers of CNN to obtain excellent results. The effectiveness of the proposed method was tested on the UC-Merced dataset. Experimental results demonstrate that the proposed classification method achieved an accuracy rate of 98.76% and 99.29% in 50% and 80% training experiment, respectively.

Keywords CNN · Feature reduction · ReliefF algorithm · Image recognition · UC-Merced dataset

Fatih Özyurt
ozyurtfatih@gmail.com

[1] Department of Informatics, Firat University, Elazig, Turkey

Published online: 14 December 2019

1 Introduction

Over the past few decades, remote sensing has undergone dramatic changes in the spatial resolution of the image and increases in the acquisition rate. Innovations in the field of computer technology and increasing spatial resolution offer new opportunities to improve image detection and remote sensing, enabling the development of new approaches. The classification of land use from remote sensing images is vital for the monitoring and management of human development in terms of scope and scale. Moreover, automatic classification through machine learning is crucial in order to understand the Earth's ever-changing surface, including anthropogenic changes. The automatic land cover classification of high-resolution remote sensing images is highly desirable but poses many difficulties due to the wide variety of volumes and various daily collected satellite and weather images. This is further complicated by a wide range of features and scales related to geographically diverse human areas on earth.

Conventional image feature extraction and image analysis techniques were guided by hand-crafted methods which typically benefited from common computer vision features extraction methods such as scale-invariant feature transformation (SIFT) [1], histograms of oriented gradients (HOG) [2] and bag of visual words (BoVW) [3]. At the same time, various features offering advantages and disadvantages have also been used [4–6]. All of these approaches succeeded in varying degrees to list only some of the wide range of image classification models such as regular photography, medical imaging and remote sensing. However, CNNs and deep learning have witnessed a revival in the past few years, promising a conceptual development in the early 1980s [7]. Deep learning techniques such as Caffe framework [8], AlexNet [9], VGG-16 [10], VGG-19 [11], GoogleNet [12], ResNet [13] and SqueezeNet [14] have recently displayed the highest performance for various machine learning tasks.

The present study focuses on feature extraction using six fused CNN architectures as feature extractors and proposes a feature reduction using the ReliefF algorithm to obtain important features of images. The proposed structure aims to aggregate the FC layer of Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet architectures by using the feature ranking ability to reach a higher classification accuracy. The proposed method also employs the transfer learning strategy to overcome overfitting and reduce training time. Pretrained CNN structures were preferred in many existing studies in the literature [15–20]. To evaluate the proposed framework, UC-Merced (UCM) Land Use dataset [3] was used. The proposed method was compared with several recent approaches and achieved state-of-the-art performance. There are four major contribution objectives in the present study.

1. Using pretrained architectures such as Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet as feature extractors.
2. Obtaining the most efficient features by combining features from the last FC layers of the six pretrained CNN architecture.
3. Obtaining best-ranked features by using the ReliefF algorithm, and using these features in SVM classification.

4. Obtaining superior performance compared with the state-of-the-art results by eliminating unnecessary features.

The rest of this article is organized as follows: Section 2 introduces the materials and methods. Section 3 describes the proposed method in detail. Section 4 presents the experimental results of the proposed method. Section 5 concludes the study.

2 Materials and methods

In this section, the UC-Merced Land Use dataset is introduced first. Secondly, I was given a brief description of the previously trained six CNN architectures. Finally, ReliefF algorithm is presented.

2.1 Dataset

The UC-Merced Land Use dataset [3] contains aerial images and is widely used in various remote sensing applications. It is a high-resolution dataset containing 2100 different images (256×256 pixel) divided into 21 unique classes which consist of agricultural, airplane, baseball-diamond, beach, buildings, chaparral, dense-residential, forest, freeway, golf-course, harbor, intersection, medium-residential, mobile-home-park, overpass, parking-lot, river, runway, sparse-residential, storage-tank, tennis-court classes. Each land use class consists of 100 images with a 1 ft spatial resolution and 256×256 pixels. The UC-Merced dataset has several overlapping classes, such as medium, sparse and dense residues, and differs only in the density of structures that make the dataset difficult to classify. Sample images are shown in Fig. 1.

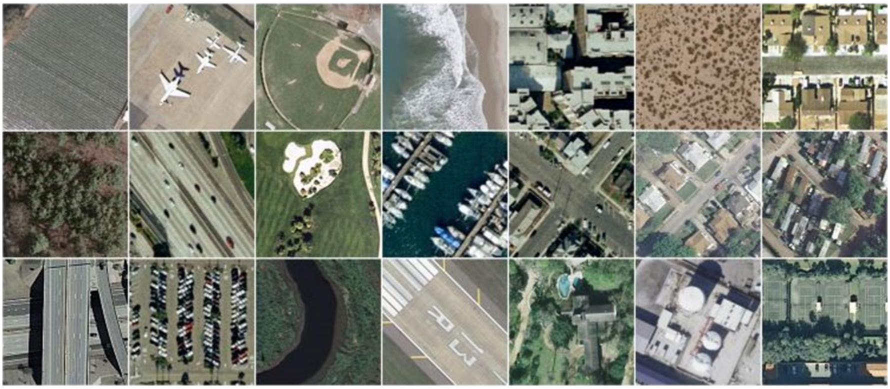


Fig. 1 Sample images of UCM dataset 21 land cover classes

2.2 Convolutional neural networks

Convolutional neural network has been widely used in image processing problems since the ImageNet competition in 2012. CNN is used in image classification, image recognition, object tracking problems and displays high performance rates [21–23]. In a convolution layer, the feature maps of the previous layer become learnable kernels and use the activation function to generate the output feature map. Each output map can combine convolutions with multiple input maps. It is generally formulated as in Eq. 1.

$$x_j^{\ell} = f\left(\sum_{i \in M_j} x_i^{\ell-1} * k_{ij}^{\ell} + b_j^{\ell} \right) \tag{1}$$

In Eq. 1, M_j represents a selection of the input map. If output map j and map k both sum over input map i, then the kernels applied to map i are different for output maps j and k. Detailed information about the mathematical representation of CNN architecture can be examined from the relevant Ref. [24]. The CNN architectures used in the present study are described below.

2.2.1 Alexnet

Designed using deep learning algorithms, AlexNet became widely known around the world in 2012 when it won the ImageNet competition. Later, it was published in an article titled “ImageNet Classification with Deep Convolutional Networks” [9]. This architecture reduced the computer image recognition error rate from 26.2% to 15.4%. Alexnet architecture consists of five convolution layers, pooling layers and three FC layers. The graphical representation of AlexNet is given in Fig. 2.

2.2.2 VGG16

Introduced by the University of Oxford Visual Geometry Group in ILSVRC-2014 competition, Vgg16 deep learning algorithm is a network consisting of 13 convolution layers and three fully connected layers in order to obtain better performances.

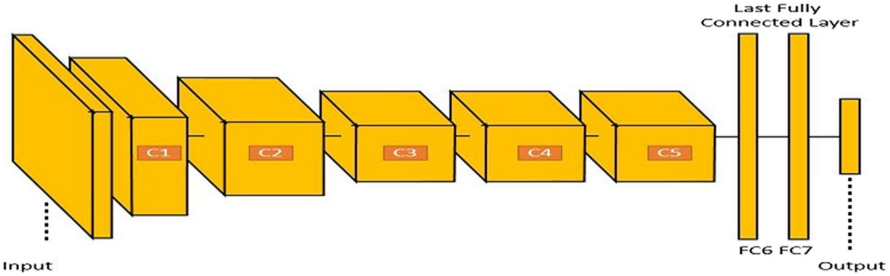


Fig. 2 Graphical representation of AlexNet

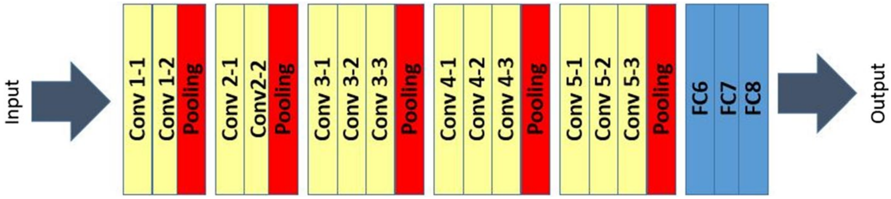


Fig. 3 Graphical representation of VGG16

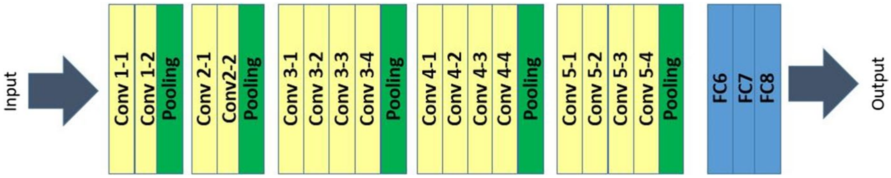


Fig. 4 Graphical representation of VGG19

It contains 41 layers including Max pool, fully connected layer, Relu layer, Dropout layer and Softmax layer. The image in the input layer has a size of 224×224×3. The last layer is the classification layer. It is a deep learning algorithm that achieved an accuracy rate of 89% in the ImageNet database [10]. The graphical representation of VGG16 is given in Fig. 3.

2.2.3 VGG19

Introduced by the University of Oxford Visual Geometry Group following Vgg16, Vgg19 deep learning algorithm is a network consisting of 16 convolution layers and three fully connected layers. There are 41 layers including Max pool, fully connected layer, Relu layer, Dropout layer and Softmax layer. Similar to Vgg16, the input layer has a size of 224×224×3. The last layer is the classification layer. It achieved an accuracy rate of 88% in the ImageNet database [11]. The graphical representation of VGG19 is given in Fig. 4.

2.2.4 GoogleNet

GoogleNet [12] is a complex architecture due to inception modules within its structure. It has 22 layers and won the ImageNet competition with an error rate of 5.7% in 2014. It can be generally considered as one of the first CNN architectures to have avoided stacking convolutional and pooling layers in a consecutive manner. Additionally, it occupies an important position in terms of memory and power users because stacking all layers and adding various filters will require time-consuming calculations and bring about higher memory costs. Graphical representation of inception modules in GoogleNet is given in Fig. 5.

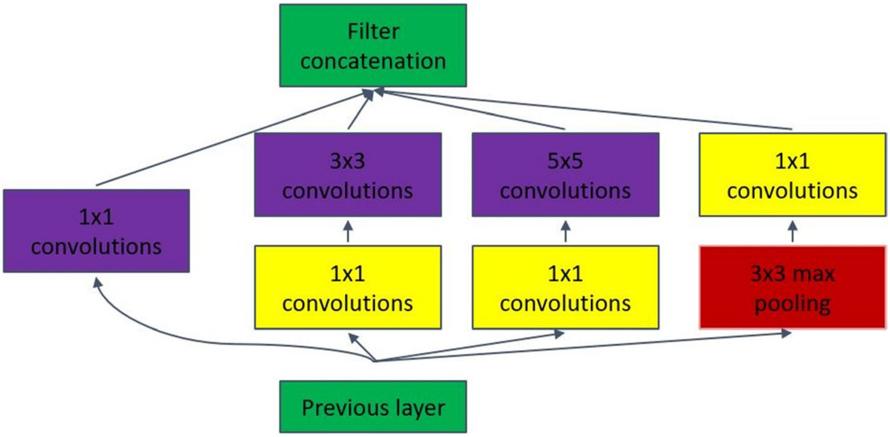


Fig. 5 Graphical representation of inception modules in GoogleNet

2.2.5 ResNet

ResNet [13] is a deeper architecture compared to all the existing architectures and consists of 152 layers. In addition, considering that people often display an error rate of 5 to 10% depending on their abilities and skills, it won the ImageNet 2015 competition with an error rate of 3.6%. It consists of residual blocks. The graphical representation of ResNet is given in Fig. 6.

2.2.6 SqueezeNet

SqueezeNet CNN architecture was introduced by Iandola et al. [14] in 2016. It has an independent convolution layer (conv1), eight fire modules (fire2–9) and the last convolution layer (conv10). While SqueezeNet performs or exceeds the first (top-1) and five highest accuracies of AlexNet, it yields a 50×decrease in model size compared to AlexNet. The graphical representation of SqueezeNet is given in Fig. 7.

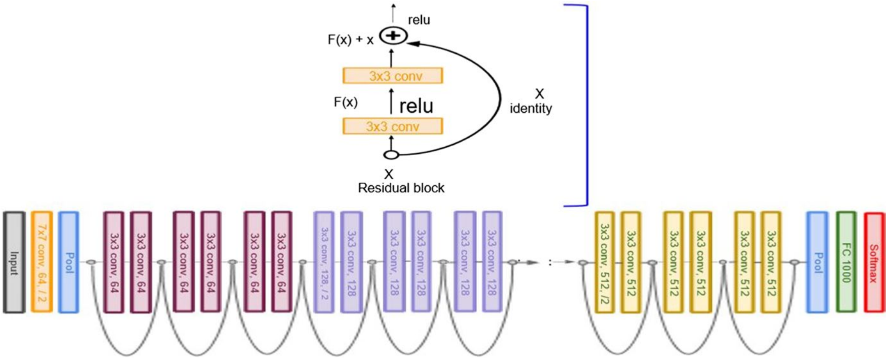


Fig. 6 Graphical representation of residual blocks in ResNet

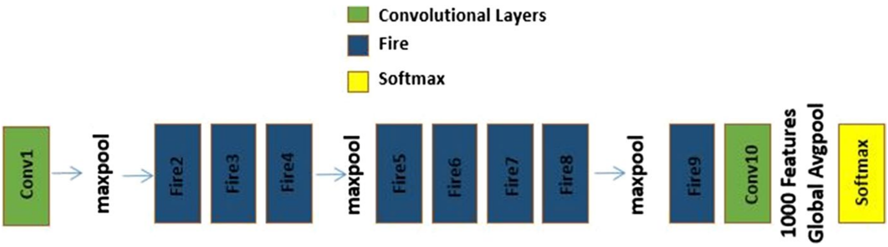


Fig. 7 Graphical representation of SqueezeNet

2.3 ReliefF algorithm

Introduced by Kira and Rendell [25] in 1992, ReliefF is one of the most successful features filtering algorithms. In 1994, the algorithm was redeveloped by Kononenko [26] for multi-class problems. Feature selection can be successfully completed with the help of this algorithm. The ReliefF algorithm offers high efficiency and does not limit the characteristics of the data types. The ReliefF algorithm helps solve multiple classes of problems by selecting the nearest neighboring samples from each sample in different categories.

ReliefF finds the weights of predictors when y is a multi-class categorical variable. The algorithm punishes predictors that give different values to neighbors of the same class and rewards those that give different values to the neighbors of different classes [27]. The formula of the ReliefF algorithm can be expressed as in Eq. 2.

$$W_f^{i+1} = W_f^i + \sum_{C\neq \text{class}(x)} \frac{\frac{p(x)}{1-p(\text{class}(x))}\sum_{j=1}^{k} \text{diff}(x, Mj(x))}{m * k} - \sum_{j=1}^{k} \text{diff}_f\left(x, H_j, (x)\right)/(m * k) \tag{2}$$

diff_f () represents feature distance between two samples. $H_j(x)$ represents the neighbor samples from the sample x. $M_j(x)$ represents the neighbor samples from neighbors of different classes. $p(x)$ represents the probability of class.

2.4 Support vector machine

Support vector machines (SVM) are machine learning algorithms based on convex optimization that work according to the structural risk minimization principle. Since the algorithm does not require any combined distribution function information, it is independent of distribution [28]. SVM was developed by Vapnik for the solution of pattern recognition and classification problems [29].

As shown in the figure, although there is a hyperplane classifier for separating both classes, there are classes that reach the maximum distance known as the optimum separating hyperplane. SVM aims to create an optimal hyperplane space using training examples. The SVM process for a binary classification problem is described as follows [30].

Now consider a binary classification problem with data points x_i; $I = 1$; 2; …, n., belong to class 1 or class 2, and their associated labels $y_i = \pm 1$, with the following decision function

$$f(x) = \mathrm{sign}(w \cdot x + b).$$

where x_{i}, w and b are training point, weight and bias vectors, respectively. The margin is the distance from the hyperplane to the nearest point for both the circle and triangle classes of data points, as shown in Fig. 8 [31]. This margin is also the optimal hyperplane that maximizes the margin. The maximization of the margin formulation is given by Eq. 3.

$$M\,\mathrm{arg}\,in = \frac{2}{\|w\|} \tag{3}$$

All data points, which are a member of that class, have to confirm the condition of $w \cdot x_i + b > 0$. y_i outputs are set of labeled x_i input data points. If y_i is 1, the x_i input data point is a member of that class. If y_i is -1, the input data point is not a member of that class. It can be given as Eq. 4.

$$\begin{aligned} w \cdot x_i + b &\geq 1, \quad \forall y_i = 1 \\ w \cdot x_i + b &\leq -1, \, \forall y_i = -1 \end{aligned} \tag{4}$$

$\|w\|$ is the norm of the normal plane w called the weight vector. Thus, the inverse distance of the samples closest to the hyperplane has to be equal to the norm of the weight vector to which it is. Based on this theorem, the hyperplane that best separates the training examples is given in Eq. 5.

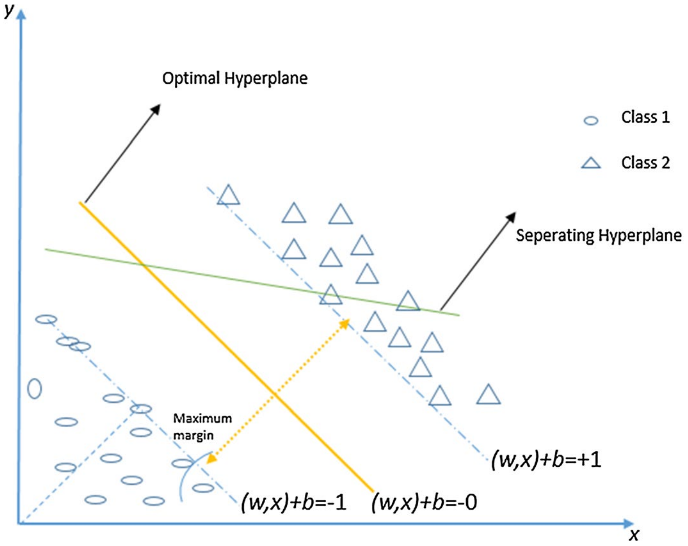


Fig. 8 Basic SVM illustration

$$\text{minimize } \frac{\|w\|^2}{2} \text{ subject to } \forall_i,\ y_i(w \cdot x_i + b) \geq 1 \tag{5}$$

To find the optimal separation hyperplane in the SVM, Lagrange multipliers are used to solve the quadratic optimization problem. Lagrange multipliers convert the smallest problem into a dual problem, allowing the problem to be solved more easily [32]. The Lagrange multipliers can be given in Eq. 6:

$$\text{For } \forall_i,\ a_i \geq 1,\ L(w, a_i, b) = \left(\|w\|^2/2\right) - \sum_{i=1}^{K} a_i(y_i(w.x_i + b) - 1) \tag{6}$$

3 The proposed method

The method proposed in the present study consists of two experimental studies and 16 steps. Firstly, the structure of Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet architectures was analyzed. Features were obtained from the last FC layer of these CNN architectures. These features were given to the SVM classifier to measure classification performance. The steps of the first experimental study were given between S***tep-2 and Step-7***. Then, the features obtained from a total of six architectures were combined into a single feature vector in ***Step-8***. Totally 6000 features were obtained at this step. In ***Step-9*** and ***Step-10***, firstly ReliefF algorithm was applied then 1000 features with the highest prediction capability were divided into 100 sub-features. The steps of the second experimental study were given between ***Step-11*** and ***Step-15***. In the second experimental study, feature subsets of 100, 200, 300, 400 and 500 were given to the SVM classifier, respectively. In the final ***Step-16***, classification performance evaluation was performed. Thus, a total of 6000 features were reduced to 500 features with the help of the ReliefF algorithm. The highest

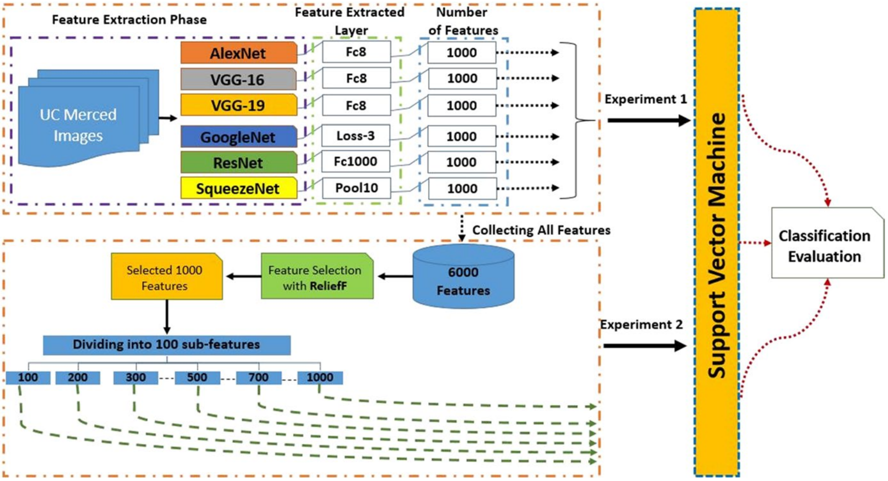


Fig. 9 Diagram of the proposed method

classification performance was found in 500 feature selection based on the obtained classification performance. The scheme of the proposed method is shown in Fig. 9. All the steps of the proposed study are given below.

Data Collection

Step-1: Original UC Merced image dataset ← Obtaining UC Merced Land Use data set [3]

Experimental Study-1

Step-2: Feature Extraction with AlexNet and, Classification← *extraction of features in the FC8 layer of AlexNet architecture and giving to SVM classifier.*

Step-3: Feature Extraction with VGG-16 and, Classification← *extraction of features in the FC8 layer of VGG-16 architecture and giving to SVM classifier.*

Step-4: Feature Extraction with VGG-19 and, Classification← *extraction of features in the FC8 layer of VGG-19 architecture and giving to SVM classifier.*

Step-5: Feature Extraction with GoogleNet and, Classification← *extraction of features in the Loss-3 layer of GoogleNet architecture and giving to SVM classifier.*

Step-6: Feature Extraction with ResNet and, Classification← *extraction of features in the FC1000 layer of ResNet architecture and giving to SVM classifier.*

Step-7: Feature Extraction with SqueezeNet and, Classification← *extraction of features in the Pool10 layer of SqueezeNet architecture and giving to SVM classifier.*

Feature Concatenation

Step-8: Feature Concatenation ← *Features obtained from* ***Step 2, step 3, step 4, step 5, step 6, step*** *7 were combined to obtain a total of 6000 features.*

Feature Selection

Step-9: Feature Selection← *Applying the ReliefF algorithm to obtaining best ranked 1000 features.*

Step-10: Feature Selection← *Obtained 1000 features with the highest prediction capability were divided into 100 sub-features.*

Experimental Study-2

Step-11: 100 Feature Subset Evaluation← *100 features with the highest prediction capability were given to SVM classifier*

Step-12: 200 Feature Subset Evaluation← *200 features with the highest prediction capability were given to SVM classifier*

Step-13: 300 Feature Subset Evaluation← *300 features with the highest prediction capability were given to SVM classifier*

Step-14: 400 Feature Subset Evaluation← *400 features with the highest prediction capability were given to SVM classifier*

Step-15: 500 Feature Subset Evaluation← *500 features with the highest prediction capability were given to SVM classifier*

Step-16: Classification Evaluation← *Evaluating classification performance by divided feature subsets.*

The bar chart representation of the 100 most efficient features obtained by the ReliefF feature selection algorithm applied to total of 6000 features is shown in Fig. 10.

The ReliefF algorithm determines the weight for each feature. These weights indicate how much the relevant features belong to the relevant class.

Other parameter selections of the CNN models used in the present study are given in Table 1. Mini-batch value can be defined as more than one input process on a model at the same time [33]. In the present study, the mini-batch value was selected as 32. The computer hardware used in the present study was limited, which led to problems in classical CNN architecture training. However, the pre-trained CNN architectures as feature extractors eliminated the need for a high-capacity computer hardware.

4 Experimental results and discussion

The experimental study was compiled using MATLAB (R2018b) software on a 64-bit Windows 10 operating system. The computer has an NVIDIA GeForce 2 GB graphics card, Intel © i7-4510U 2.6 GHz processor and an 8 GB RAM. The studies were divided into two categories.

4.1 Experimental results and analysis

In the first experimental study, the features of 2100 images belonging to 21 classes in the UC-Merced dataset were extracted using CNN architectures. Thousand features from the last FC layer of each architecture were classified using the SVM classifier. Test accuracy values were calculated by dividing the dataset into two separate experimental sets as 50% and 80% training rates. The results of this study are given in Table 2.

According to the graph shown in Fig. 11, the highest performance was displayed in the experimental study where 80% of training was performed using ResNet architecture at an accuracy rate of 96.9%. In another experimental study in which the same architecture had 50% training, an accuracy rate of 93.62% was achieved. The performance of ResNet architecture was followed by SqueezeNet, which, despite a 50 × reduction in the model size compared to AlexNet, performed better compared to the AlexNet architecture. In addition, although the number of convolution layers in VGG-19 architecture is more than VGG-16, lower performance rates have been obtained. The lowest performance rate was achieved when 50% of training was conducted with GoogleNet architecture. It is understood that the complexity of the CNN architecture does not imply a better feature.

In the second experimental study, the ReliefF algorithm was applied to a total of 6000 features obtained from six CNN architectures. The first 1000 features with the highest prediction capability were divided into sub-feature sets of 100 parts. A total of 100, 200, 300, 400, 500 and 600 features were given as an input of the SVM classifier. Test accuracy values were calculated by dividing

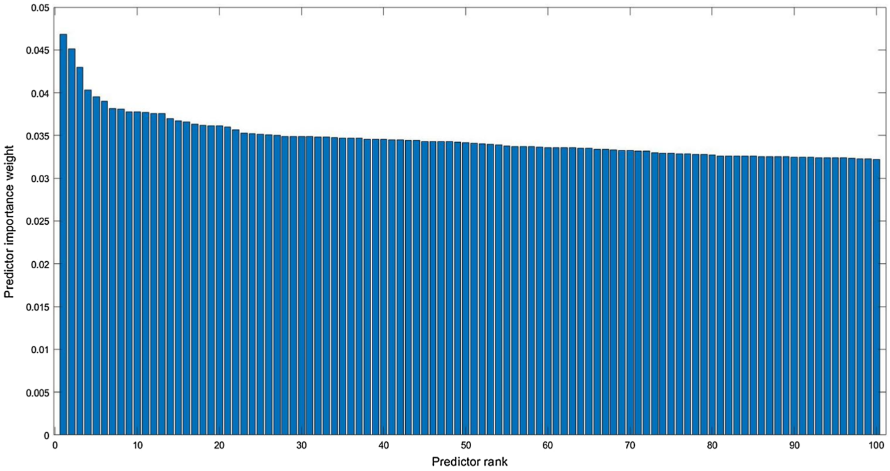


Fig. 10 Sample bar chart graph of the best 100 feature weights

the dataset into two separate experimental sets as 50% and 80% training rates. It was observed that the highest classification performance was displayed by 500 features. The data were used for two different experimental studies: 50% training, 50% testing, 80% training and 20% testing. An accuracy rate of 98.76% and 99.29% was obtained in 50% and 80% training rates, respectively. The results of the experiments are shown in Table 3.

When Fig. 12 is analyzed, it can be observed that the performance rate increased rapidly when moving from 100 feature sets to 200 feature sets. This rate increased slightly in each subset of features. Considering the lowest feature subset and highest performance rate criteria, it can be stated that optimal performance rates were displayed by 500 features. The success rates obtained with 100 sets of features reduced by the ReliefF algorithm are higher than all individual performance rates in Table 2. The small number of features does not always affect the classification performance negatively. This clearly demonstrates the consistency of the proposed method.

As a result, the combined dataset with ReliefF feature selection algorithm achieved the highest performance in 500 features. Confusion matrices of these studies are shown in Figs. 13 and 14, respectively.

In the confusion matrices shown in Figs. 13 and 14, 1 represents agricultural class; 2 represents airplane class; 3 represents baseball-diamond class; 4 represents beach class; 5 represents buildings class; 6 represents chaparral class; 7 represents dense-residential class; 8 represents forest class; 9 represents freeway

Table 1 Common parameter values of CNN architectures

Software used	Momentum	Decay	Beta	Mini-batch	Learning rate
Matlab	0.90	1E−5	–	32	0.0001

Table 2 Classification of features obtained using CNN architecture

Pretrained CNN architecture	Method	Number of features used (last FC layer)	Training ratio	
			Test accuracy rate (%)	
			50%	80%
Alexnet	Alexnet-SVM	1000	90.10	93.81
VGG-16	VGG-16-SVM	1000	90.10	94.05
VGG-19	VGG-19-SVM	1000	89.71	93.10
GoogleNet	GoogleNet-SVM	1000	89.62	94.29
ResNet	ResNet-SVM	1000	93.62	96.90
SqueezeNet	SqueezeNet-SVM	1000	92.38	95.24

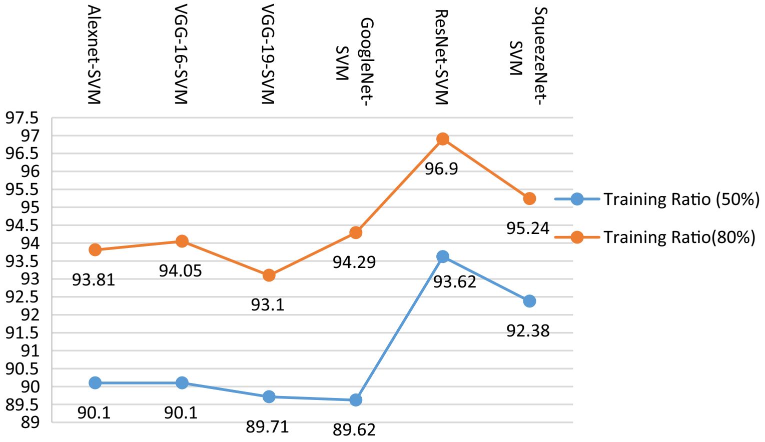


Fig. 11 Classification performance of architectures

Table 3 The results of the sub-feature sets divided into 100 parts

Method	Feature subset	Training ratio	
		Test accuracy rate (%)	
		50%	80%
CNN-Relief-SVM	100	95.81	97.86
	200	97.71	98.80
	300	98.10	99.05
	400	98.57	99.29
	500	98.76	99.29

class; 10 represents golf-course class; 11 represents harbor class; 12 represents intersection class; 13 represents medium-residential class; 14 represents mobile-home-park class; 15 represents overpass class; 16 represents parking-lot class; 17

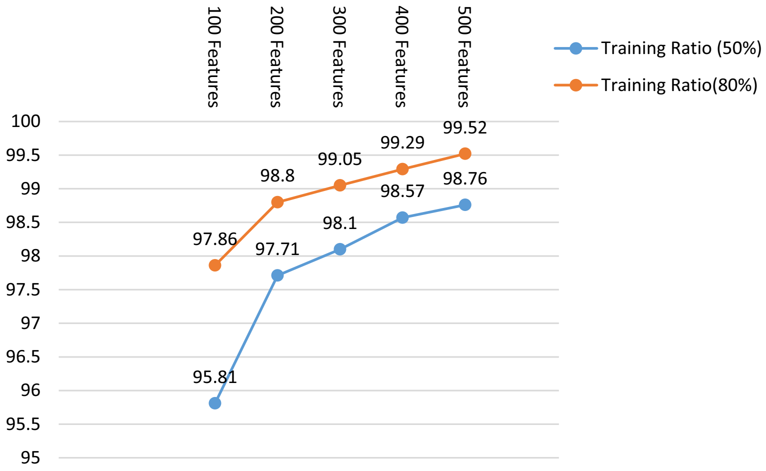
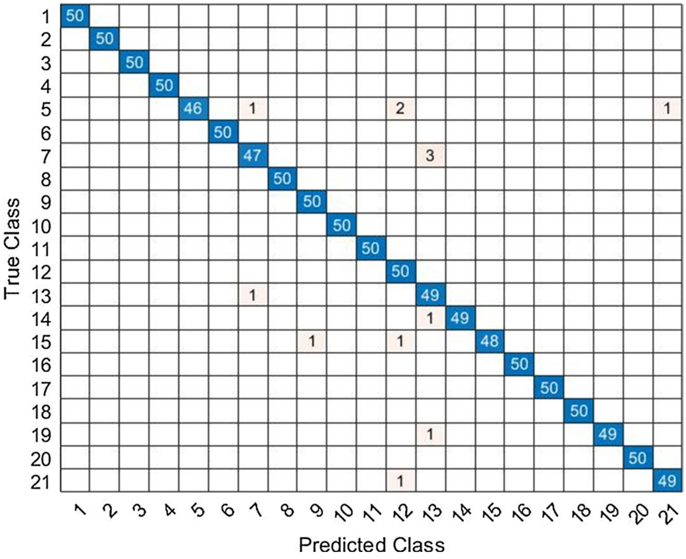


Fig. 12 Accuracy rate of CNN-Relief-SVM method


True Class
Predicted Class

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	50																				
2		50																			
3			50																		
4				50																	
5					46		1					2									1
6						50															
7							47						3								
8								50													
9									50												
10										50											
11											50										
12												50									
13							1						49								
14													1	49							
15									1			1			48						
16																50					
17																	50				
18																		50			
19													1						49		
20																				50	
21												1									49

Fig. 13 Confusion matrix of 50% trained data (98.76%)

represents river class; 18 represents runway class; 19 represents sparse-residential class; 20 represents storage-tanks class; and 21 represents tennis-court class.

In order to make comparisons with different methods using the same dataset, we presented the general accuracy rates and the confusion matrices in Figs. 13 and 14. The overall accuracy ratio is expressed as the ratio of accurately classified images to the total number of images. An NxN table summarizing how successful the estimates of a classification model are is called the confusion matrix. It is an

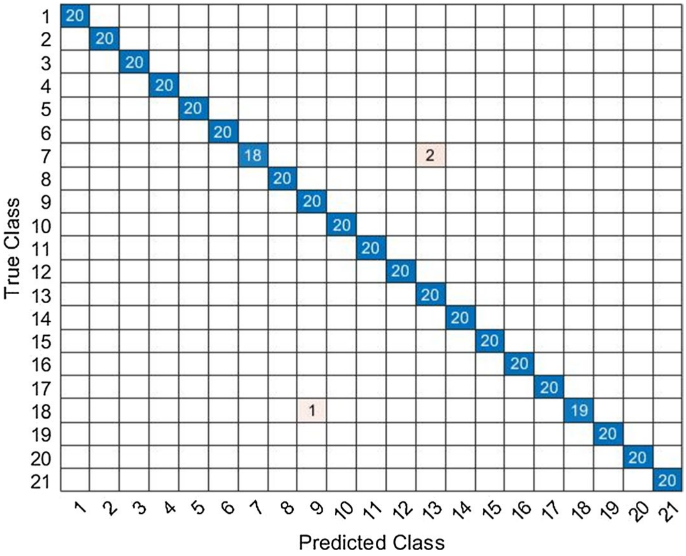


Fig. 14 Confusion matrix of 80% trained data (99.29%)

informative table expressing the correlation between the label and the classification of the model. In order to express the general accuracy of the study, the training data were randomly selected. The classification of the data was repeated ten times to obtain convincing average accuracy and standard deviation.

4.2 Comparison with state-of-the-art methods

The proposed method was compared with other state-of-the-art methods using the same dataset. In general, existing studies have so far been conducted at 80% training or 50% training rates.

As given in Table 4, the proposed method displayed the highest performance rate. This shows the effectiveness of the proposed method. The proposed method produced the most advanced performance with accuracy rates of 99.29% and 98.76%, using 80% and 50% labeled samples per class, respectively. The proposed method gave better results than the second highest accuracy of 99% and 97.37% reported in [47] in this dataset. [47] proposed a new CNN model from end to end by integrating global content features (GCFs) and local object-level features (LOFs). In another study close to the proposed study, two pretrained CNN algorithms were combined with the two-stream fusion [46] technique and given to ELM classifier. In fact, this study is similar to the proposed method. However, in the proposed article, the features obtained from the six pretrained CNN architecture were combined and then the redundant features were reduced. This has enabled the proposed method to yield higher performance rates.

Table 4 Accuracy rate of proposed method

Method	Training ratio	
	Accuracy (%)	
	50%	80%
MCMI [34]	–	88.20
Unsupervised feature learning [35]	–	81.67 ± 1.23
Gradient boosting CNNs [36]	–	94.53
LGF [37]	–	95.48
SSF-AlexNet [38]	–	92.43
Multifeature concatenation [39]	–	92.38 ± 0.62
CaffeNet [40]	93.98 ± 0.67	95.02 ± 0.81
GoogLeNet [40]	92.70 ± 0.60	94.31 ± 0.89
VGG16 [40]	94.14 ± 0.69	95.21 ± 1.20
Fine-tuned GoogLeNet [41]	–	97.10
Deep CNN transfer [42]	–	98.49
salM^{3}LBP-CLM [43]	94.21 ± 0.75	95.75 ± 0.80
TEX-Net-LF [44]	95.89 ± 0.37	96.62 ± 0.49
Fusion by addition [45]	–	97.42 ± 1.79
Two-stream fusion [46]	96.97 ± 0.75	98.02 ± 1.03
CNN-GCFs-LOFs [47]	97.37 ± 0.44	99 ± 0.35
CNN-Relief-SVM	98.76 ± 0.56	99.29 ± 0.26

5 Conclusion

In the present study, the most popular pretrained architectures such as Alexnet, VGG16, VGG19, GoogleNet, ResNet and SqueezeNet were used as feature extractors. First, the features obtained from the last FC layers of these architectures were given to the SVM classifier separately. Then, separate classification performances of each architecture were measured. Later, the features obtained from the latest FC layers of these architectures were combined to obtain a total of 6000 features. ReliefF algorithm was applied to these features. ReliefF algorithm lists the weights of features that affect classification performance. The first 100, 200, 300, 400 and 500 features with the highest feature weight were classified using SVM, which is a hybrid method called CNN-Relief-SVM method. In this study, a new method was proposed to improve classification accuracy by using fewer features. In this method, required parameter settings for data training in classical CNN architectures were eliminated. This method provides a fast and effective method for the classification and recognition of remote sensing images, but further research is needed to use features derived from architectures efficiently. Experimental results clearly indicate that the CNN-Relief-SVM method performs better compared to other state-of-the-art methods. In the future, optimization-based methods can be developed to extract efficient features from many architectures at the same time.

Funding There is no funding source for this article.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest related to this paper.

References

1. Kobayashi T (2014) Dirichlet-based histogram feature transform for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3278–3285
2. Negrel R, Picard D, Gosselin PH (2014) Evaluation of second-order visual features for land-use classification. In: 2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI), pp 1–5. IEEE
3. Yang Y, Newsam S (2010) Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp 270–279. ACM
4. Hu F, Xia GS, Wang Z, Huang X, Zhang L, Sun H (2015) Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification. IEEE J Sel Top Appl Earth Obs Remote Sens 8:5
5. Chen C, Zhang B, Su H, Li W, Wang L (2016) Land-use scene classification using multi-scale completed local binary patterns. SIViP 10(4):745–752
6. Chen S, Tian Y (2014) Pyramid of spatial relatons for scene-level land use classification. IEEE Trans Geosci Remote Sens 53(4):1947–1957
7. Fukushima K (1980) Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36(4):193–202
8. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp 675–678. ACM
9. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1097–1105
10. Qassim H, Verma A, Feinzimer D (2018) Compressed residual-VGG16 CNN model for big data places image recognition. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp 169–175. IEEE
11. Mateen M, Wen J, Song S, Huang Z (2019) Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry 11(1):1
12. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–9
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
14. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and < 0.5 MB model size. *arXiv preprint* arXiv:1602.07360
15. Kutlu H, Avcı E (2019) A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks. Sensors 19(9):1992
16. Kutlu H, Avci E, Özyurt F (2019) White blood cells detection and classification based on regional convolutional neural networks. Med Hypotheses 135:109472
17. Penatti OA, Nogueira K, Dos Santos JA (2015) 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, pp 44–51
18. Özyurt F, Sert E, Avci E, Dogantekin E (2019) Brain tumor detection based on convolutional neural network with neutrosophic expert maximum fuzzy sure entropy. Measurement 147:106830

19. Özyurt F, Sert E, Avci D (2019) An expert system for brain tumor detection: fuzzy C-means with super resolution and convolutional neural network with extreme learning machine. Med Hypotheses 134:109433
20. Özyurt F (2020) A fused CNN model for WBC detection with MRMR feature selection and extreme learning machine. Soft Comput 83:1–10
21. Understanding of Convolutional Neural Network (CNN)—Deep Learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. Accessed 19 May 2019
22. Sert E, Özyurt F, Doğantekin A (2019) A new approach for brain tumor diagnosis system: single image super resolution based maximum fuzzy entropy segmentation and convolutional neural network. Med Hypotheses 133:109413
23. Lv X, Ming D, Chen Y, Wang M (2019) Very high resolution remote sensing image classification with SEEDS-CNN and scale effect analysis for superpixel CNN classification. Int J Remote Sens 40(2):506–531
24. Bouvrie J (2006) Notes on Convolutional Neural Networks [Online]. Available: http://cogprints.org/5869/1/cnn_tutorial.pdf
25. Kira K, Rendell LA (1992) The feature selection problem: traditional methods and a new algorithm. Aaaı 2:129–134
26. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. In: Bergadano F, De Raedt L (eds) Machine Learning: ECML-94. ECML 1994. Lecture Notes on Computer Science (Lecture Notes in Artificial Intelligence), vol 784. Springer, Berlin, Heidelberg
27. Tuncer T, Ertam F (2019) Neighborhood component analysis and reliefF based survival recognition methods for Hepatocellular carcinoma. Physica A Stat Mech Appl 12:123143
28. Soman KP, Loganathan R, Ajay V (2009) Machine learning with SVM and other kernel methods. PHI Learning Pvt. Ltd., New Delhi
29. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
30. Vapnik V, Levin E, Le Cun Y (1994) Measuring the VC-dimension of a learning machine. Neural Comput 6(5):851–876
31. Kindermann J, Paaß G, Leopold E (2001) Error correcting codes with optimized Kullback-Leibler distances for text categorization. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, pp 266–276
32. Pierna JF, Baeten V, Renier AM, Cogdill RP, Dardenne P (2004) Combination of support vector machines (SVM) and near-infrared (NIR) imaging spectroscopy for the detection of meat and bone meal (MBM) in compound feeds. J Chemom A J Chemom Soc 18(7–8):341–349
33. Yang Z, Wang C, Zhang Z, Li J (2019) Mini-batch algorithms with online step size. Knowl Based Syst 165:228–240
34. Ren J, Jiang X, Yuan J (2015) Learning LBP structure by maximizing the conditional mutual information. Pattern Recogn 48(10):3180–3190
35. Cheriyadat AM (2014) Unsupervised feature learning for aerial scene classification. IEEE Trans Geosci Remote Sens 52:439–451
36. Zhang F, Du B, Zhang L (2016) Scene classification via a gradient boosting random convolutional network framework. IEEE Trans Geosci Remote Sens 54:1793–1802
37. Zou J, Li W, Chen C, Du Q (2016) Scene classification using local and global features with collaborative representation fusion. Inf Sci 348:209–226
38. Liu Y, Zhong Y, Fei F, Zhu Q, Qin Q (2018) Scene classification based on a deep random-scale stretched convolutional neural network. Remote Sens 10:444
39. Shao W, Yang W, Xia GS, Liu G (2013) A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization. In: *International Conference on Computer Vision Systems*. Springer, Berlin, Heidelberg, pp 324–333
40. Xia GS, Hu J, Hu F, Shi B, Bai X, Zhong Y, Lu X (2017) AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans Geosci Remote Sens 55(7):3965–3981
41. Castelluccio M, Poggi G, Sansone C, Verdoliva L (2015) Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint* arXiv:1508.00092
42. Hu F, Xia GS, Hu J, Zhang L (2015) Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens 7(11):14680–14707
43. Bian X, Chen C, Tian L, Du Q (2017) Fusing local and global features for high-resolution scene classification. IEEE J Sel Top Appl Earth Obs Remote Sens 10(6):2889–2901

44. Anwer RM, Khan FS, van de Weijer J, Molinier M, Laaksonen J (2018) Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. ISPRS J Photogramm Remote Sens 138:74–85
45. Chaib S, Liu H, Gu Y, Yao H (2017) Deep feature fusion for VHR remote sensing scene classification. IEEE Trans Geosci Remote Sens 55(8):4775–4784
46. Yu Y, Liu F (2018) A two-stream deep fusion framework for high-resolution aerial scene classification. Comput Intell Neurosci. https://doi.org/10.1155/2018/8639367
47. Zeng D, Chen S, Chen B, Li S (2018) Improving remote sensing scene classification by integrating global-context and local-object features. Remote Sens 10(5):734

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.