Machine Learning with Remote Sensing Image Datasets

Biserka Petrovska
Ministry of Defense, Republic of North Macedonia
E-mail: biserka.petrovska@morm.gov.mk

Tatjana Atanasova-Pacemska, Natasa Stojkovik, Aleksandra Stojanova and Mirjana Kocaleva
Faculty of Computer Science, University "Goce Delcev," Republic of North Macedonia
E-mail: tatjana.pacemska@ugd.edu.mk; natasa.stojkovik@ugd.edu.mk; aleksandra.stojanova@ugd.edu.mk; mirjana.kocaleva@ugd.edu.mk

Keywords: machine learning, remote sensing, convolutional neural networks, transfer learning, feature extraction, fine-tuning

Received: September 1, 2020

Computer vision, as a part of machine learning, gains significant attention from researches nowadays. Aerial scene classification is a prominent chapter of computer vision with a vast application: military, surveillance and security, environment monitoring, detection of geospatial objects, etc. There are several publicly available remote sensing image datasets, which enable the deployment of various aerial scene classification algorithms. In our article, we use transfer learning from pre-trained deep Convolutional Neural Networks (CNN) within remote sensing image classification. Neural networks utilized in our research are high-dimensional previously trained CNN on ImageNet dataset. Transfer learning can be performed through feature extraction or fine-tuning. We proposed a two-stream feature extraction method and afterward image classification through a handcrafted classifier. Fine-tuning was performed with adaptive learning rates and a regularization method label smoothing. The proposed transfer learning techniques were validated on two remote sensing image datasets: WHU RS datasets and AID dataset. Our proposed method obtained competitive results compared to state-of-the-art methods.

Povzetek: Metoda prenesenega učenja je uporabljena za analizo posnetkov iz zraka na nekaj referenčnih bazah.

1 Introduction

Scene classification is a process of assigning a semantic label to remote sensing (RS) images [1, 2]. It is one of the crucial tasks in aerial image understanding. Aerial scene classification is possible due to the existence of several RS images datasets collected from satellites, aerial systems, and unmanned aerial vehicles (UAV). Remote sensing image classification has located its utilization in many fields: military, traffic observation, and disaster monitoring [3, 4]. The problem of aerial scene classification is complex because the composition of remote sensing images is compound, and it is rich in features: space and texture. This is the reason for developing numerous scene classification methods.

Remote sensing image classification methods that rely on feature extraction can be categorized in one of the following groups: methods that use low-level image features, methods that use mid-level image features and methods that utilize high-level image representation. Methods using low-level image features operate on aerial scene classification with low-level visual descriptors: spectral, textural, structural, etc. Scale Invariant Feature Transform (SIFT) is a local descriptor that simulates local fluctuation of structures in remote sensing images [5]. Statistical and global allocation of certain image characteristics: color [6] and texture data [7] are utilized by other descriptors. Different color and texture descriptors, like color histograms and local binary pattern (LBP) descriptors, are comparatively analyzed in [8]. Remote sensing classification in [9] is performed by compound-feature figures of 6 different types of descriptors: SIFT, radiometric features, Grey Level Co-Occurrence Matrix (GLCM), Gaussian wavelet features, shape features, and Gabor filters, with varying spatial resolution. Other descriptors used by researches are the orientation difference descriptor [10], and the Enhanced Gabor Texture Descriptor (EGTD) [11]. For aerial scene classification, authors in [12] use completed local binary patterns with multi-scales (MS-CLBP) and achieved state-of-the-art-results compared to other methods based on low-level image features.

Mid-level image features methods try to represent aerial images with a statistical representation of a high degree obtained from the extracted local image features. The first step within these methods is to extract local image features from local patches employing descriptors like SIFT or color histograms. The second step is to encode those features to obtain a mid-level representation for remote sensing images. A widely used mid-level method is bag-of-visual-words (BoVW) [13]. The first step of BoVW is to extract features with SIFT from local image patches [14], and afterward, to learn so-called visual dictionary or visual codebook, that is a vocabulary of visual words. In aerial scene classification tasks, the basic BoVW technique can be combined with various local descriptors [14]: GIST, SIFT, color histogram, LBP. Another mid-level method relies on a sparse coding method [15] where low level extracted features such as structural, spectral, and textural are encoded. Improvement of the classification accuracy can be obtained with Principal Component Analysis (PCA) which enables dimensionality reduction of extracted features before fusing them in compound-representatives, or with methods such as the Improved Fisher Vector (IFK) [16] and Vectors of Locally Aggregated Tensors (VLAT) [17]. In the literature can be found improved models of BoVW like spatial pyramid co-occurrence kernel (SPCK) [18], which integrates the absolute and relative spatial data. This method combines concepts of spatial pyramid match kernel (SPM) [19] and spatial co-occurrence kernel (SCK) [13]. In [20] pyramid-of-spatial-relations (PSR) model is presented, which includes absolute and relative spatial connections of local low-level features.

The third group of feature extraction methods for image classification relies on high-level vision information. The latest techniques that include high-level features based on CNN learning have shown significant improvement of classification accuracies compared to older low-level and mid-level image features methods. High-level methods can acquire more abstract and discriminative semantic representations, which guides in improved classification performance. Feature extraction with deep neural networks, previously trained on ImageNet data set [21], results in significant performance for aerial scene classification [22]. Remote sensing image classification accuracy achieved with GoogleNet [23] can be improved with an input strategy of multi-ratio for multi-view CNN learning. Multi-scale image features are extracted from the last convolutional layer of CNN [24] and then encoded with BoVW [25], Vector of Locally Aggregated Descriptors (VLAD) [26] and Improved Fisher Kernel (IFK) [16] to compose the final image representation. Nogueira et al. [27] extracted global features from CNN architectures and guided them to a classifier. In all of the examples mentioned above, the global or local extracted features were obtained from CNNs previously trained on massive data sets like ImageNet, formed of natural images. Extracted features were utilized for remote sensing image classification.

Another method of transfer learning is the fine-tuning of CNN weights. It is a technique where the original classification layer of the pre-trained CNN (usually softmax layer) is replaced with a new one, which contains a number of nodes equal to a number of classes of the target dataset. Altered CNN is trained with a random initialization of new layers, but the remaining layers begin with the pre-trained weights. Compared to a neural network training with random weight initialization, fine-tuning achieves a better minimum of the lost function. Authors in [28] achieved significant performance improvement by fine-tuning a pre-trained CNN. They experimented with AlexNet [29] and obtained a better outcome for semantic segmentation. Also, there are several papers in the remote sensing community [30], [31], that surveyed the benefits of fine-tuning CNNs. A comparison between CNN trained from scratch, and fine-tuned one showed the advantages of using aerial scene data [31]. The fine-tuning method could be useful for the classification of hyperspectral images [30]. Fine-tuning weights of pre-trained CNNs results in the extraction of better scene features [32]. This transfer learning technique, performed on neural networks previously trained on the ImageNet dataset, results in good classification accuracy on remote sensing image data sets [24], [27]. Our previous work, [53], [54], showed that transfer learning techniques, feature extraction as well as fine-tuning, are superb methods for aerial scene classification.

Despite the two transfer learning methods described above, the other alternative is to train CNN from scratch, i.e., with random initialization of network weights. This solution shows low classification accuracy for small-scale aerial scene datasets [27]. Full network training of CaffeNet and GoogLeNet resulted in poor classification results for the UC-Merced dataset [13] and the WHU-RS19 dataset [33]. But, full CNN training using large-scale datasets like AID [34] and NWPU-RESISC45 [35] has obtained good results.

In this paper, we evaluate miscellaneous CNN models on resolving the task of high-resolution aerial scene classification. We utilize convolutional neural networks pre-trained on ImageNet data set with a twofold purpose: like feature extractors and for fine-tuning on particular remote sensing datasets. When we use pre-trained CNN as feature extractors, we try to form better features for aerial imagery. Thus, we acquire activations from different CNN layers: the average pooling layer, the last, and one of the intermediate convolutional layers. In order to enable the fusion of features from convolutional layers with ones from average pooling layers, the feature dimensionality reduction method is utilized on those from convolutional layers. Compound features of the image scenes are processed by a linear classifier to determine image classes.

In the second experimental setup, we explore the fine-tuning of network weights on the remote sensing imagery. We trained CNNs with adaptive learning rates: linear decay schedule and cyclical learning rates and assessed if they are appropriate for fine-tuning of pre-trained CNN on aerial scene imagery. In order to achieve classification accuracy comparable to state-of-the-art methods, we included label smoothing as a regularization technique and assessed its impact on the experimental results.

The main contributions of this paper are (1) evaluation of transfer learning techniques with various CNN models on two remote sensing image datasets, (2) analysis of the impact of fused features obtained by concatenation of activations from different pre-trained CNN layers on classification accuracy, (3) assessment of the influence of adaptive learning rates at the fine-tuning method from the aspect of classification accuracy, and (4) the proposed transfer learning techniques are compared to state-of-the-art methods and provide a baseline for aerial imagery classification.

The remainder of this article is organized as follows. In Section 2, the methodologies used for transfer learning from CNN are presented. Experimental results obtained from the examined remote sensing images classification methods are presented in Section 3. Discussion of impact factors on our method's results, as well as summarization and conclusion of the paper, is given in Section 4.

2 Materials and methods

This section of the article gives a short description of the pre-trained CNNs used for transfer learning: InceptionV3, ResNet50, Xception, and DenseNet121. Following that, we introduce the PCA for dimensionality reduction, linear decay scheduler, and cyclical learning rates as methods for transfer learning. Next, we present the two publicly available data sets: WHU-RS19 and AID included in our experiments. Finally, the utilized experimental setup and the evaluation metrics are given.

2.1 Convolutional neural networks

ResNet won the classification task part of ILSVRC-2015. ResNet is a deep CNN that can have up to 152 layers [49]. It is similar to the VGG model because it contains mostly 3x3 filters, but the number of filters is smaller, and CNN is simpler [49]. Deep learning architectures can have high training error and vanishing gradient problem. The solution to the vanishing gradient problem is including a residual module in the neural network. The deep learning residual module, as shown in Figure.1, has a short connection between the input and the output.

The first inception based network was named Inception-v1 or GoogleNet [50].

In GoogleNet architecture, inception modules are included, and thus the number of learning parameters is decreased. The original inception module, Figure.2, has a pooling layer and convolutional layers with dimensions 1x1, 3x3, and 5x5. Module output is got by concatenating the outputs of these layers. Inception based networks relay on the detail that the correlation within the image pixels is local. The number of learning parameters is reduced based on the local correlations. Inception-v3 [37] is the third iteration of inception based networks. It contains three different types of inception modules: type 1, got by dividing into smaller convolutions; type 2, got by dividing into asymmetric convolutions; and type 3, that was included to improve representations with high dimensions.

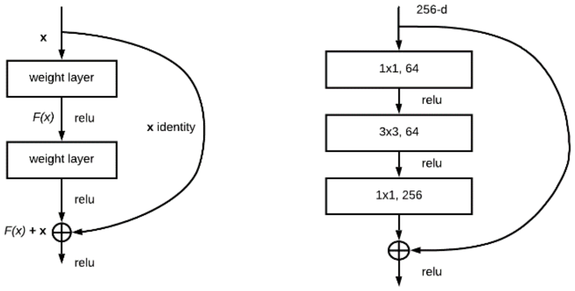


Figure 1: Residual block (left) and “bottleneck” block (right) of ResNet.

Another deep CNN, which is similar to Inception, is Xception. In the Xception, the inception module is replaced with depth-wise separable convolutional layers [51]. This CNN is a cluster of depth-wise separable convolutional layers with shortcut connections. A depthwise separable convolution is separated into two phases. The first phase is a spatial convolution applied separately on each input channel, so-called depthwise convolution. After that, pointwise convolution follows, which is 1x1 convolution for conveying the output of depthwise convolution output channels to a new channel space.

Dense Convolutional Network (DenseNet) [52], enables highest data flow between network layers, by attaching layers to each other in a feed-forward manner. The only condition for such connections is the layers to have corresponding dimensions of feature maps. The input for each layer are the feature maps of the preceding layers, and its own feature maps are carried into all layers ahead, as their input. Opposite to ResNets, here the authors [52] fuse features with concatenation, but don't add together the features to lead them afterward into the following layer. This neural network got its name after the dense connectivity pattern, Dense Convolutional Network (DenseNet). The dense pattern suggests that there is no need to relearn redundant feature maps, which leads DenseNet to have a smaller number of parameters than other deep CNN.

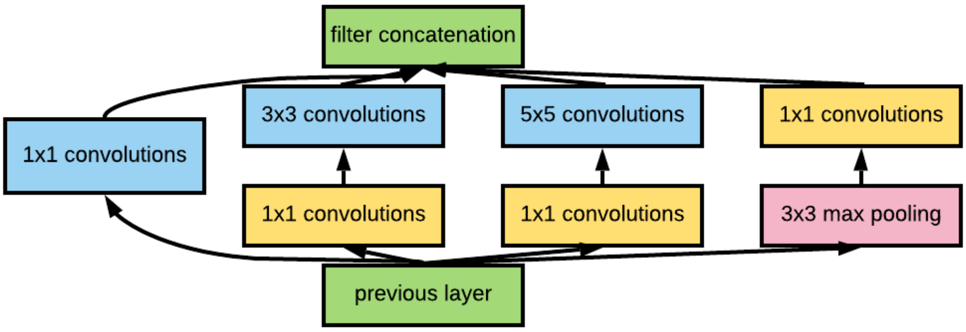


Figure 2: The architecture of a basic inception module.

2.2 Principal Component Analysis

In our experiments, we used Principal Component Analysis (PCA) as a dimensionality-reduction technique. It establishes a new group of the basis of view, and then project the data from the original representation to a representation with fewer dimensions. The new dimensions are orthogonal to each other, independent and ordered, depending on the variance of data they contain. The first principal component is the one with the highest variance. The new data matrix consists of n data points with k features for each of them:

$$[new\ data]_{k*n} = [top\ k\ eigenvectors]_{k*m}[original\ data]_{m*n} \quad (1)$$

The covariance matrix is symmetric. The variance of every dimension is on the main diagonal, and the covariances of dimensions are placed elsewhere. PCA is a dimensionality reduction method that spreads out data to have high variance within a fewer number of dimensions.

2.3 Adaptive learning rates

The most crucial hyperparameters for CNN training are initial learning rate, number of training epochs, learning rate schedule, and regularization method (L2, dropout). The invariable learning rate for network training might be a reasonable choice in some instances, but more often, an adaptive learning rate is more beneficial. When training CNN, we are trying to find global, local minima, or only a part of the loss surface with adequate low loss. If we train the network with a constant but large learning rate, we can't reach the desired valley of loss terrain. But if we adapt (decrease) our learning rate, the neural network can descend into more optimal parts of the loss landscape.

In our proposed fine-tuning method, we use a linear decay schedule, which decays our learning rate to zero at the end of the network training. The learning rate α in every training epoch is given with:

$$\alpha = \alpha_I * (1 - \frac{E}{E_{max}}) \quad (2)$$

where αI is the learning rate at the beginning of training, E is the number of the current epoch, and Emax is the overall number of epochs.

Cyclical Learning Rates (CLR) are another form of adaptive learning rates. In this case, there is no need to determine the optimal initial learning rate and schedule for the learning rate when we train CNN [36]. When the network is trained with learning rate schedules, the learning rate is being continuously reduced, but CLR allows the learning rate to oscillate among pre-defined limits. The network training with CLR convergences faster with fewer hyperparameter updates.

Authors in [36] define a few CLR policies: triangular shown in Figure 3, triangular2, and exponential range policy. The triangular policy, as can be seen in Figure 3, is a triangular pattern: the learning rate oscillates linearly between the fixed lower limit and the upper limit. Triangular2 policy looks similar to triangular policy, except that the upper limit of a learning rate is twice lower after every cycle. As a result of this, triangular2 policy training is more stable. The exponential range policy encompasses exponential declination of a maximum learning rate.

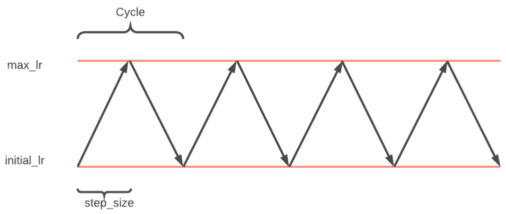


Figure 3: Cyclical learning rate with a triangular policy model.

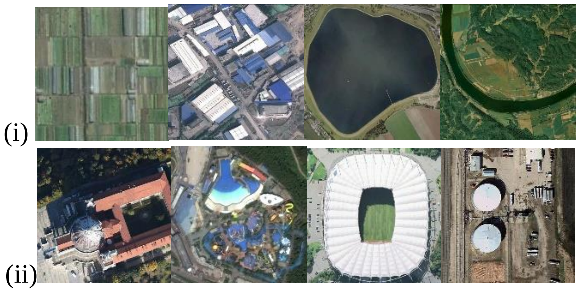


Figure 4: Some images of different classes from (i) WHU RS and (ii) AID data set.

2.4 Remote sensing datasets

We test our proposed transfer learning techniques on two common aerial scene data sets; the WHU RS data set [43] and the aerial image dataset (AID) [34].

The WHU-RS data set [43] is selected from Google Earth imagery, and the images are collected from all over the world. There are 19 image classes, with at least 50 images per class, entirely 1005 images. Image dimensions are 600x600 pixels. WHU-RS data set has been extensively used in experimental studies of remote sensing classification tasks. Image classes in the WHU-RS data set are airport, beach, bridge, commercial, desert, farmland, football field, forest, industrial, meadow, mountain, park, parking, pond, port, railway station, residential, river, and viaduct.

The aerial image dataset (AID) has approximately 10,000 remote sensing images assigned to 30 classes: airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks, and viaduct. Image dimensions are 600×600 pixels with a pixel resolution of half a meter. Images are obtained from Google Earth imagery. They are picked from various world regions at different times of the year and climate conditions: mostly from China, Japan, Europe, and the United States.

2.5 Experimental setup and evaluation metrics

The first proposed transfer learning method was based on feature extraction. We involved four different pre-trained CNNs and extracted features from three various layers of each of them. For ResNet50 the layers were: bn4f_branch2c, the last convolutional layer, and average pooling layer; for InceptionV3: mixed_8, mixed_10 and average pooling layer; for Xception: block14_sepconv1_act, block14_sepconv2_act, and average pooling layer; and for DenseNet121 the layers were: conv4_block24_concat, the last convolutional layer, and average pooling layer. Before feature extraction, data set images were resized and pre-processed according to the demands of each pre-trained CNN. Data augmentation was applied to the images of the training set. Five patches of each training image were made with rotation, shifting, shearing, zooming, and flipping. The feature extraction was performed for the WHU-RS data set, for 60%/40% and 40%/60% training/test split ratio. The splits are random and without stratification. We included feature fusion to improve the classification performance of the proposed method. At first, image features were extracted from two various layers of two CNNs, so that one layer was the average pooling layer, and the other was some of the aforementioned convolutional layers. Then, we applied PCA transformation on convolutional layer features. Before the features are concatenated, we performed L2 normalization on PCA transformed convolutional layer features and the average pooling layer features. After the features fusion, a linear Support Vector Machine SVM classifier is trained with compound features.

The SVM is a classifier described by a separating hyperplane. SVM model has four hyperparameters: type of kernel, the influence of regularization, gamma, and margin. The kernel can be linear, exponential, polynomial, etc. We used a linear kernel in our experimental setup. The prediction of a classifier for linear kernel is given with

$$\mathrm{f(x)} = \mathrm{B}(0) + \mathrm{sum}(a_i * (\mathrm{x}, \mathrm{x}_i)) \qquad (3)$$

The output of the classifier is acquired with the dot product of the input (x) and each of the support vectors (xi). The model computes the inner products of each input vector (x) with all support vectors in training images. The learning algorithm determines coefficients B(0) and ai from the training data.

In our proposed feature extraction method, we tuned the regularization parameter. During SVM optimization, the regularization parameter regulates to what extent to take into consideration the misclassification of each training image.

Our proposed fine-tuning method for aerial scene classification, as a form of transfer learning, is carried out with adaptive learning rates, as well as label smoothing. Label smoothing is a regularization method that fights against overfitting and leads to a better generalization of the CNN model. It is expectable our model to overfit because we use pre-trained CNN with high dimensionality and fine-tune them with a data set that has only a couple of thousands of images. Label smoothing [37] magnifies classification accuracy evaluating the cost function with “soft” labels from the data set (weighted sum of the labels with equal distribution) instead of “hard” labels. When we apply label smoothing with parameter α, we reduce the cost function between the ‘smoothed’ labels y_k^LS and the network outcome pk, smoothed labels are as follows:

$$\mathrm{y}_\mathrm{k}^\mathrm{LS} = \mathrm{y}_k\,(1-\alpha) + \alpha/\mathrm{K} \qquad (4)$$

where the real labels are y_k, and K is the number of classes. Label smoothing was applied only to the training images. In-place data augmentation was used for training images as well. In the simulation scenario, we included four pre-trained CNN: ResNet50, InceptionV3, Xception, and DenseNet121, and images of the target data set were resized according to the requirements of each CNN. The fine-tuning method was applied to the AID data set, and the experiments were performed with 50%/50% and 20%/80% train/test data split ratios. To prepare pre-trained CNN for fine-tuning, we removed from each network the layers after the average pooling layer. On top of this, a new CNN head was constructed by adding a fully connected layer, dropout layer, and softmax classifier.

We started fine-tuning with warming-up the new CNN head. We warmed new network layers with an RMSprop optimizer and a constant learning rate. The fine-tuning continued with Stochastic Gradient Descent (SGD), and training was performed on all network layers. Different simulations scenarios were carried out with a linear decay schedule and cyclical learning rates with triangular policy. When it comes to the linear decay learning rate, the initial learning rate was selected relatively small, 1-2 orders of magnitudes lower than the initial learning rate of the originally trained CNN. Cyclical learning rates oscillated between the maximum and minimum bound with the optimal learning rate somewhere in between. The step size is equal to 4 or 8 times the number of training iterations in the epoch, and the number of epochs is chosen to contain integer of cycles.

In our paper, we use the following evaluation metrics: Overall Accuracy OA (classification accuracy) and confusion matrix. OA is the ratio between the number of correctly classified test images and the total number of test images. It is always lower than 1 (lower than 100%). The confusion matrix is a table that represents the partial accuracy of each image class. This graphical display shows the errors of every single different class and confusion between the classes. Here the columns appear for the predicted classes, and the rows appear for the real classes. Better classification accuracy leads to higher values of the main diagonal of the confusion matrix and lower values for other entries. To check the reliability of the results, all cases are repeated ten times (five times for fine-tuning method). After that, the mean value and the standard deviation (SD) for each experiment are calculated.

3 Results

3.1 Classification of WHU-RS dataset

The feature extraction transfer learning method was evaluated on the WHU-RS data set. Accuracy of SVM classification of compound features from the average pooling layer and PCA transformed convolutional layer features is shown in Table 1.

Table 2 presents a comparative analysis of the proposed feature extraction method to competitive classification methods. It can be concluded that feature fusion with PCA transformation is a technique that achieves state-of-the-art classification accuracies. Under a training ratio of 40% of the WHU-RS data set, this method outperforms all the other classification methods.

Figure 5 and Figure 6 show the confusion matrices without normalization obtained from the classification of WHU-RS data set under 60% training data with InceptionV3 mixed_8 (PCA) and ResNet50 average pooling, and under 40% training data with DenseNet121 conv5_block16_concat (PCA) and ResNet50 average pooling.

3.2 Classification of AID dataset

The experimental results of the fine-tuning method for classification of the AID dataset are displayed in Table 3. As can be seen from Table 3, the linear decay scheduler gives better classification results for a 50%/50% train/test split ratio for ResNet50 and InceptionV3. Cyclical learning rate works better for Xception and DenseNet121. For 20%/80% train/test split ratio linear decay scheduler is a better option for ResNet50, Xception and DenseNet121, and cyclical learning rates for InceptionV3.

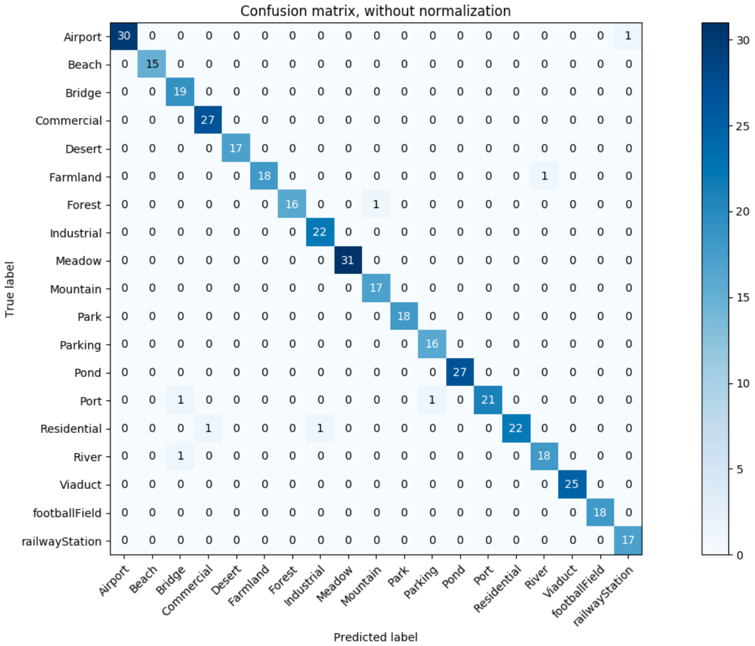
Table 4 is a comparative display of our fine-tuning method with other state-of-the-art techniques. Our method achieved the best classification results on the AID dataset

Method	60% training ratio	40% training ratio
ResNet50 last conv layer (PCA) and InceptionV3 average pool	98.26	95.02
ResNet50 last conv layer (PCA) and Xception average pool	97.62	96.52
InceptionV3 mixed_10 (PCA) and ResNet50 average pool	96.27	95.85
InceptionV3 mixed_8 (PCA) and ResNet50 average pool	98.01	98.67
InceptionV3 mixed_10 (PCA) and Xception average pool	96.77	96.02
InceptionV3 mixed_8 (PCA) and Xception average pool	98.01	96.35
DenseNet121 conv5_block16_concat (PCA) and ResNet50 average pool	98.76	98.34
DenseNet121 conv4_block24_concat (PCA) and ResNet50 average pool	96.77	96.52

Table 1: Classification accuracy of feature extraction method with WHU-RS data set.

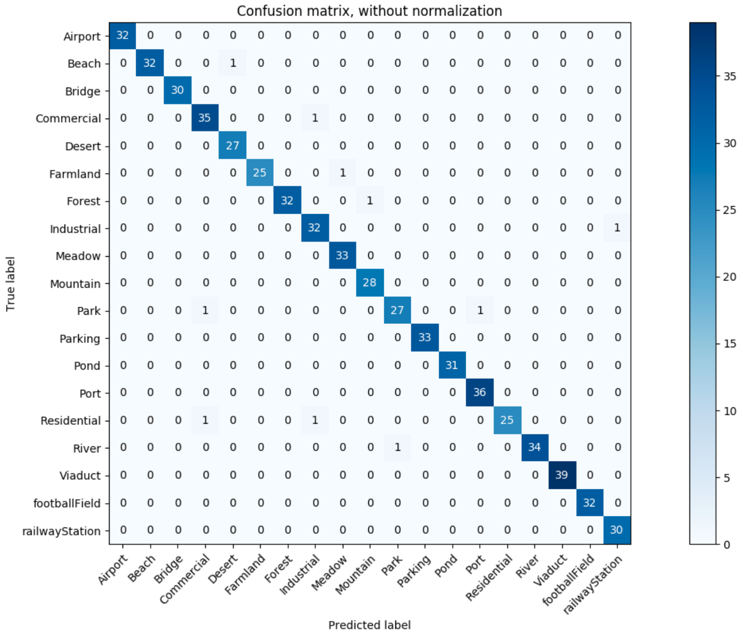
Method	60% of WHU-RS data set as a training set	40% of WHU-RS data set as a training set
Bag of SIFT [20]	85.52 ± 1.23	/
Multi Scale Completed LBP + BoVW [44]	89.29 ± 1.30	/
GoogLeNet [34]	94.71 ± 1.33	93.12 ± 0.82
VGG-VD-16 [34]	96.05 ± 0.91	95.44 ± 0.60
CaffeNet [34]	96.24 ± 0.56	95.11 ± 1.20
salM^{3}LBP-CLM [45]	96.38 ± 0.82	95.35 ± 0.76
TEX-Network-LF [46]	96.62 ± 0.49	95.89 ± 0.37
InceptionV3 mixed_8 (PCA) and ResNet50 average pool (Ours)	**98.13 ± 0.51**	/
DCA by concatenation [47]	98.70 ± 0.22	97.61 ± 0.36
Addition with saliency detection [48]	98.92 ± 0.52	98.23 ± 0.56
DenseNet121 conv5_block16_concat (PCA) and ResNet50 average pool (Ours)	**/**	**98.26 ± 0.40**

Table 2: Classification accuracy (%) and standard deviation of the state-of-the-art methods with WHU-RS data set.



	Airport	Beach	Bridge	Commercial	Desert	Farmland	Forest	Industrial	Meadow	Mountain	Park	Parking	Pond	Port	Residential	River	Viaduct	footballField	railwayStation
Airport	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Beach	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bridge	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Commercial	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Desert	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Farmland	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	1	0	0	0
Forest	0	0	0	0	0	0	16	0	0	1	0	0	0	0	0	0	0	0	0
Industrial	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0
Meadow	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0
Mountain	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
Park	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0
Parking	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0
Pond	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0
Port	0	0	1	0	0	0	0	0	0	0	0	1	0	21	0	0	0	0	0
Residential	0	0	0	1	0	0	0	1	0	0	0	0	0	0	22	0	0	0	0
River	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0
Viaduct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
footballField	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0
railwayStation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17

Figure 5: Confusion matrix of the feature extraction method under 60% training data of WHU-RS data set.



	Airport	Beach	Bridge	Commercial	Desert	Farmland	Forest	Industrial	Meadow	Mountain	Park	Parking	Pond	Port	Residential	River	Viaduct	footballField	railwayStation
Airport	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Beach	0	32	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bridge	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Commercial	0	0	0	35	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Desert	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Farmland	0	0	0	0	0	25	0	0	1	0	0	0	0	0	0	0	0	0	0
Forest	0	0	0	0	0	0	32	0	0	1	0	0	0	0	0	0	0	0	0
Industrial	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0	1
Meadow	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	0
Mountain	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0
Park	0	0	0	1	0	0	0	0	0	0	27	0	0	1	0	0	0	0	0
Parking	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0
Pond	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0
Port	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0
Residential	0	0	0	1	0	0	0	1	0	0	0	0	0	0	25	0	0	0	0
River	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	34	0	0	0
Viaduct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0
footballField	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0
railwayStation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30

Figure 6: Confusion matrix of the feature extraction method under 40% training data of WHU-RS data set.

for InceptionV3, under 50% training data with a linear decay scheduler, and under 20% training data with cyclical learning rates. As can be concluded from Table 4, some methods outperformed the proposed fine-tuning technique, like EfficientNet-B3-aux [38]. Authors in [38] used the fine-tuning of the EfficientNet-B3 network with auxiliary classifier. The explanation for better classification results might be that the network mentioned above has achieved better top-1 classification accuracy on ImageNet data set than CNNs we utilized in our experimental setup.

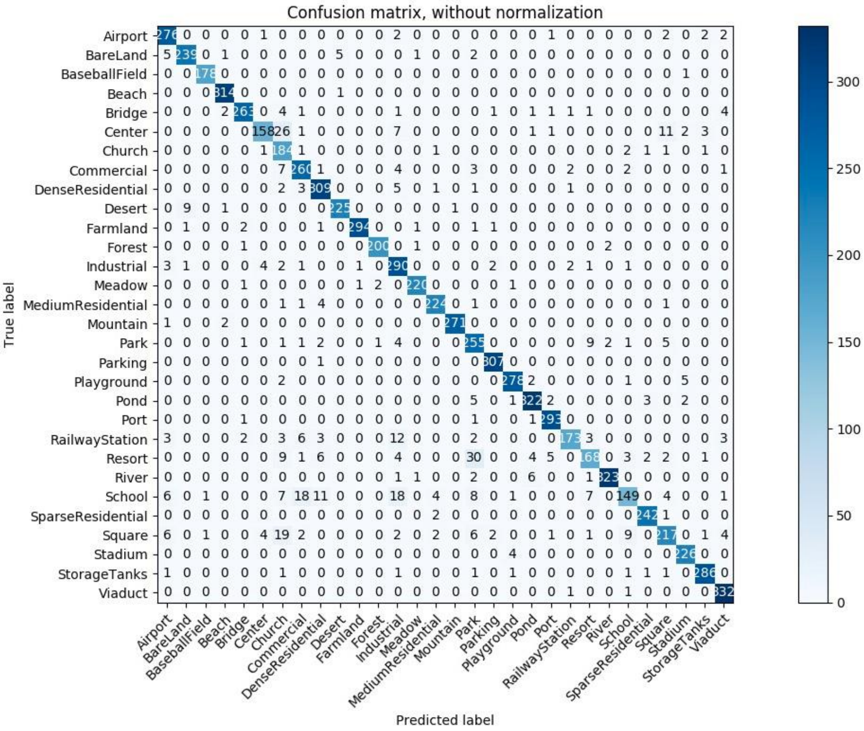
Figure 7 displays the confusion matrix of the AID dataset classification for the proposed fine-tuning method with a 20%/80% train/test split ratio for ResNet50, cyclical learning rates, and softmax classifier. The main diagonal shows the number of properly predicted test images; the other elements give misclassified test images.

Method	**50% training ratio**	**20% training ratio**
ResNet50		
Linear decay scheduler	95.62±0.15	93.06±0.16
Cyclical learning rate	95.52±0.28	92.91±0.35
InceptionV3		
Linear decay scheduler	96.41±0.23	93.7±0.33
Cyclical learning rate	95.95±0.2	93.79±0.24
Xception		
Linear decay scheduler	96.14±0.12	93.67±0.18
Cyclical learning rate	96.15±0.17	93.44±0.10
DenseNet121		
Linear decay scheduler	96.03±0.16	93.74±0.24
Cyclical learning rate	96.21±0.19	93.54±0.15

Table 3: Overall accuracy (%) and standard deviation of the fine-tuning method with the AID data set.

Method	**50% training ratio**	**20% training ratio**
CaffeNet [34]	89.53±0.31	86.86±0.46
MCNNs [55]	91.80±0.22	/
Fusion by concatenation [39]	91.87±0.36	/
TEX-Net-LF [46]	92.96±0.18	90.87±0.11
VGG-16 (fine-tuning) [40]	93.60±0.64	89.49±0.34
Multilevel fusion [56]	95.36±0.22	/
GBNet +global Feature [40]	95.48±0.25	92.20±0.23
InceptionV3-CapsNet [41]	96.32±0.12	93.79±0.13
InceptionV3 with linear decay scheduler (ours)	**96.41±0.23**	93.7±0.33
InceptionV3 with cyclical learning rate (ours)	95.95±0.2	**93.79±0.24**
EfficientNet-B3-aux [38]	96.56±0.14	94.19±0.15
GCFs + LOFs [42]	96.85±0.23	92.48±0.38

Table 4: Overall accuracy (%) and standard deviation of the fine-tuning method compared to state-of-the-art methods for the AID data set.



	Airport	BareLand	BaseballField	Beach	Bridge	Center	Church	Commercial	DenseResidential	Desert	Farmland	Forest	Industrial	Meadow	MediumResidential	Mountain	Park	Parking	Playground	Pond	Port	RailwayStation	Resort	River	School	SparseResidential	Square	Stadium	StorageTanks	Viaduct
Airport	276	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	2	2
BareLand	5	239	0	1	0	0	0	0	0	5	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
BaseballField	0	0	178	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Beach	0	0	0	314	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bridge	0	0	0	2	263	0	4	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	4
Center	0	0	0	0	0	158	26	1	0	0	0	0	7	0	0	0	0	0	0	1	1	0	0	0	0	0	11	2	3	0
Church	0	0	0	0	0	1	184	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	1	0	1	0
Commercial	0	0	0	0	0	0	7	260	1	0	0	0	4	0	0	0	3	0	0	0	0	2	0	0	2	0	0	0	0	1
DenseResidential	0	0	0	0	0	0	2	3	309	0	0	0	5	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Desert	0	9	0	1	0	0	0	0	0	225	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Farmland	0	1	0	0	2	0	0	0	1	0	294	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Forest	0	0	0	0	1	0	0	0	0	0	0	200	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Industrial	3	1	0	0	0	4	2	1	0	0	1	0	290	0	0	0	0	2	0	0	0	2	1	0	1	0	0	0	0	0
Meadow	0	0	0	0	1	0	0	0	0	0	1	2	0	220	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
MediumResidential	0	0	0	0	0	0	1	1	4	0	0	0	0	0	224	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Mountain	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	271	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Park	0	0	0	0	1	0	1	1	2	0	0	1	4	0	0	0	255	0	0	0	0	0	9	2	1	0	5	0	0	0
Parking	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	307	0	0	0	0	0	0	0	0	0	0	0	0
Playground	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	278	2	0	0	0	0	1	0	0	5	0	0
Pond	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	1	322	2	0	0	0	0	3	0	2	0	0
Port	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	293	0	0	0	0	0	0	0	0	0
RailwayStation	3	0	0	0	2	0	3	6	3	0	0	0	12	0	0	0	2	0	0	0	0	173	3	0	0	0	0	0	0	3
Resort	0	0	0	0	0	0	9	1	6	0	0	0	4	0	0	0	30	0	0	4	5	0	168	0	3	2	2	0	1	0
River	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2	0	0	6	0	0	1	323	0	0	0	0	0	0
School	6	0	1	0	0	0	7	18	11	0	0	0	18	0	4	0	8	0	1	0	0	0	7	0	149	0	4	0	0	1
SparseResidential	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	242	1	0	0	0
Square	6	0	1	0	0	4	19	2	0	0	0	0	2	0	2	0	6	2	0	0	1	0	1	0	9	0	217	0	1	4
Stadium	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	226	0	0
StorageTanks	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	286	0
Viaduct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	332

Figure 7: Confusion matrix of the fine-tuning technique under 20% training data of AID dataset for ResNet50, cyclical learning rates, and softmax classifier.

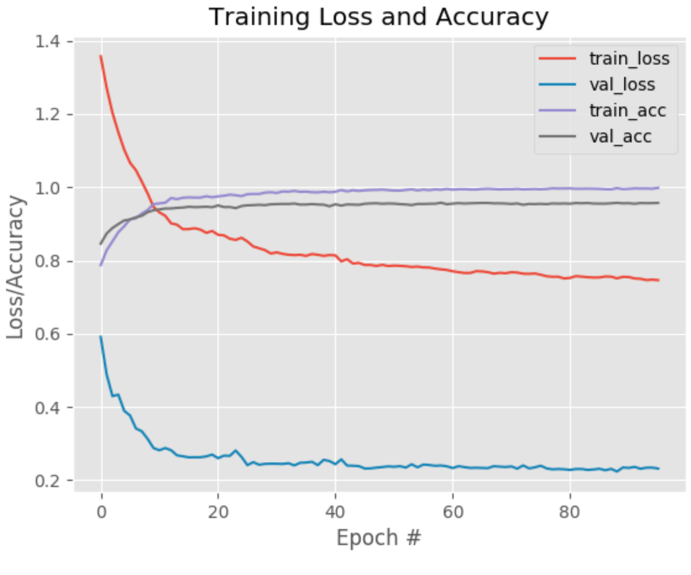


Figure 8: Training plot of the fine-tuning technique under 50% training data of AID data set for InceptionV3, cyclical learning rate, and softmax classifier.

with a 50%/50% train/test split ratio of AID data set with cyclical learning rates and softmax classifier. The plot shows the fine-tuning when all layers are "trainable" with an SGD optimizer. The plot has a characteristic shape for training with cyclical learning rates; the form of training and validation loss lines is "wavy." Because we fine-tuned the network with smoothed train labels, the training loss is higher than validation loss.

4 Discussion and conclusion

From the completed simulations and obtain results, the following valuable concepts can be summed up:

- When it comes to feature extraction method, Inception V3, and DenseNet121 are the pre-trained CNN that give the highest classification accuracies. As it is presented in Table 1, the best experimental results on the WHU-RS dataset are obtained when features from these networks' layers are fused. From Table 3 it is evident that InceptionV3 outperforms other pre-trained CNNs in transfer learning through fine-tuning for the AID data set;

- The most suitable layer for feature extraction is mixed_8 from Inception V3. It gives good classification results with ResNet50, as well as DenseNet121 average pooling layer. ResNet50 average pooling layer also gives significant classification results when it is combined with DenseNet121 convolutional layers, the last or the intermediate ones;
- For the fine-tuning method, under 50% training data ratio linear learning rate decay scheduler gives better classification results for ResNet50 and Inception V3 pre-trained networks, and cyclical learning rates are a better choice for Xception and DenseNet121. Under 20% training data ratio, learning rate decay scheduler works better for Xception and DenseNet121, and cyclical learning rates are a better choice for ResNet50 and Inception V3;
- Our proposed transfer learning methods give classification accuracies comparable to state-of-the-art techniques. The feature fusion method with PCA transformation gives the classification accuracy of 98.26 ± 0.40 under a 40% training ratio of the WHU-RS dataset, which outperforms other methods in the literature. For the fine-tuning method applied to the AID dataset, some methods obtain better experimental results compared to ours, like EfficientNet-B3-aux [38], and the reason for better classification accuracy might be the type of pre-trained CNN utilized in the scenario.

In our paper, we proposed two distinct transfer learning techniques for remote sensing image classification. The feature extraction method utilizes the concatenation of extracted features from different CNNs' layers with prior PCA transformation. The fine-tuning method includes adaptive learning rates and label smoothing. With both transfer learning methods, we have achieved significant classification results on the two datasets. The proposed feature extraction technique can be further explored with feature extraction from lower layers of pre-trained CNN, as well as with stratification of train/test data split. For future development of the fine-tuning method, we suggest including different types of pre-trained CNNs apart from the ones used in this article, like EfficientNets, and involving of learning rate finder [36] to discover optimal values for initial learning rate or limits for cyclical learning rates.

References

[1] A. Qayyum, A. S. Malik, N. M. Saad, M. Iqbal, M. F. Abdullah, and W. Rasheed, Scene classification for aerial images based on CNN using sparse coding technique. International Journal of Remote Sensing, vol.38, pp.2662–2685, 2017.

[2] J. Gan, Q. Li, Z. Zhang, and J. Wang, Two-level feature representation for aerial scene classification. IEEE Geoscience and Remote Sensing Letters, vol.13, no.11, pp.1626–1630, 2016.

[3] W. Yang, X. Yin, and G. S. Xia, Learning high-level features for satellite image classification with limited labeled samples. IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 8, pp.4472–4482, 2015.

[4] F. Huang and L. Yan, Hull vector-based incremental learning of hyperspectral remote sensing images. Journal of Applied Remote Sensing, vol.9, no.1, Article ID096022, 2015.

[5] D. G. Lowe, Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004

[6] M. J. Swain and D. H. Ballard, Color indexing. International journal of computer vision, vol. 7, no. 1, pp. 11–32, 1991

[7] V. Risojevic, and Z. Babic, Aerial image classification using structural texture similarity. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, 2011, pp. 190–195

[8] J. A. dos Santos, O. A. B. Penatti, and R. da Silva Torres, Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. in VISAPP (2), 2010, pp. 203–208

[9] B. Luo, S. Jiang, and L. Zhang, Indexing of remote sensing images with different resolutions by multiple features. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 6, no. 4, pp. 1899–1912, 2013.

[10] V. Risojevic, and Z. Babic, Orientation difference descriptor for aerial image classification. International Conference on Systems, Signals, and Image Processing (IWSSIP). IEEE, 2012, pp. 150–153

[11] V. Risojevic, and Z. Babic, Fusion of global and local descriptors for remote sensing image classification. IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 4, pp. 836–840, 2013

[12] C. Chen, B. Zhang, H. Su, W. Li, and L. Wang, Land-use scene classification using multi-scale completed local binary patterns. Signal, Image, and Video Processing, pp. 1–8, 2015

[13] Y. Yang and S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification. Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2010, pp. 270–279

[14] L. Chen, W. Yang, K. Xu, and T. Xu, Evaluation of local features for scene classification using vhr satellite images. Joint Urban Remote Sensing Event (JURSE). IEEE, 2011, pp. 385–388.

[15] G. Sheng, W. Yang, T. Xu, and H. Sun, High-resolution satellite scene classification using a sparse coding based multiple feature combination. International journal of remote sensing, vol. 33, no. 8, pp. 2395–2412, 2012

[16] F. Perronnin, J. Sanchez, and T. Mensink, Improving the fisher kernel for large-scale image classification. Proc. European Conference on Computer Vision, 2010, pp. 143–156.

[17] R. Negrel, D. Picard, and P.-H. Gosselin, Evaluation of second-order visual features for land-use classification. International Workshop on Content-Based Multimedia Indexing (CBMI). IEEE, 2014, pp. 1–5.

[18] Y. Yang and S. Newsam, Spatial pyramid co-occurrence for image classification. IEEE International Conference on Computer Vision (ICCV). IEEE, 2011, pp. 1465–1472

[19] S. Lazebnik, C. Schmid, and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2169–2178

[20] S. Chen and Y. Tian, Pyramid of spatial relations for scene-level land use classification. IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 4, pp. 1947–1957, 2015

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, pp. 1–42, April 2015.

[22] O. A. B. Penatti, K. Nogueira, and J. A. DosSantos, Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'15), pp.44–51, IEEE, Boston, Mass, USA, June 2015

[23] F. P. S. Luus, B. P. Salmon, F. VanDenBergh, and B. T. J. Maharaj, Multi-view deep learning for land-use classification. IEEE Geoscience and Remote Sensing Letters, vol.12, no.12, pp.2448– 2452, 2015.

[24] 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. 14680–14707, 2015.

[25] J. Sivic and A. Zisserman, Video google: A text retrieval approach to object matching in videos, in Proc. IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.

[26] H. J´egou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp. 1704–1716, 2012

[27] Nogueira, K.; Penatti, O.A.B.; dos Santos, J.A., Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognit. 2017, 61, 539–556

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. Computer Vision and Pattern Recognition, IEEE, 2014, pp. 580–587

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems, 2012, pp. 1106–1114

[30] J. Yue, W. Zhao, S. Mao, and H. Liu, Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. Remote Sensing Letters 6 (6) (2015) 468–477.

[31] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon, Transfer learning from deep features for remote sensing and poverty mapping. arXiv preprint arXiv:1510.00098

[32] 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

[33] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Mantre, Structural high-resolution satellite image indexing, in ISPRS TC VII Symposium-100 Years ISPRS, vol. 38, 2010, pp. 298–303

[34] G.S. Xia, J. Hu, F. Hu, B. Shi, AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification, IEEE Transactions on Geoscience and Remote Sensing, vol.55, 2017, pp. 3965-3981

[35] G. Cheng, J. Han, X. Lu, Remote Sensing Image Classification: Benchmark and State of the Art, Proceedings of the IEEE, vol.105, 2017, pp. 1865-1883

[36] L. Smith, Cyclical learning rates for Training Neural Networks. arXiv:1506.01186v6, 2017

[37] C. Szegedy, V. Vanhouck, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567v3, 2015

[38] Y. Bazi, M.M. Al Rahhal, H. Alhichri, and N. Alajlan, Simple Yet Effective Fine-Tuning of Deep CNNs Using an Auxiliary Classification Loss for Remote Sensing Scene Classification. Remote Sens. 2019, 11(24), 2908; https: //doi.org/10.3390/rs11242908

[39] G. Wang, B. Fan, S. Xiang, and C. Pan, Aggregating Rich Hierarchical Features for Scene Classification in Remote Sensing Imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2017, 10, 4104–4115.

[40] H. Sun, S. Li, X. Zheng, and X. Lu, Remote Sensing Scene Classification by Gated Bidirectional Network. IEEE Trans. Geosci. Remote Sens. 2019, 1–15.

[41] W. Zhang, P. Tang, and L. Zhao, Remote Sensing Image Scene Classification Using CNN-CapsNet. Remote Sens. 2019, 11, 494.

[42] D. Zeng, S. Chen, B. Chen, and S. Li, Improving remote sensing scene classification by integrating global-context and local-object features. Remote Sens. 2018, 10, 734

[43] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Mantre, Structural high-resolution satellite image indexing, in ISPRS TC VII Symposium-100 Years ISPRS, vol. 38, 2010, pp. 298–303

[44] L. Huang, C. Chen, W. Li, and Q. Du, Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors, Remote Sensing, vol.8, no.6, article no.483, 2016

[45] X. Bian, C. Chen, L. Tian, Q. Du, Fusing local and global features for high-resolution scene classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2017, 10, 2889–2901.

[46] R.M. Anwer, F.S. Khan, J. vandeWeijer, M. Monlinier, J. Laaksonen, Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. arXiv2017, arXiv: 1706.01171.

[47] S. Chaib, H. Liu, Y. Gu, H. Yao, Deep feature fusion for VHR remote sensing scene classification. IEEE Trans. Geosci. Remote Sens. 2017, 55, 4775–4784.

[48] Y. Yu, F. Liu, A two-stream deep fusion framework for high-resolution aerial scene classification. Comput. Intell. Neurosci. 2018, 2018, 8639367.

[49] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, arXiv:1512.03385v1, 10 Dec 2015

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, IEEE Conf. on Comput. Vision and Pattern Recognition, Boston, MA, June 2015, pp. 1–9.

[51] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, arXiv: 1610.02357v3, 4 Apr 2017

[52] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely Connected Convolutional Networks, arXiv:1608.06993v5, 28 Jan 2018

[53] Petrovska B., Zdravevski E., Lameski P., Corizzo R., Stajduhar I., Lerga J., Deep Learning for Feature Extraction in Remote Sensing: A Case-study of Aerial Scene Classification. Sensors 2020, 14, 3906

[54] Petrovska B., Atanasova-Pacemska T., Corizzo R., Mignone P., Lameski P., Zdravevski E., Aerial Scene Classification through Fine-Tuning with Adaptive Learning Rates and Label Smoothing. Appl. Sci. 2020, 10, 5792

[55] Liu Y., Huang C. Scene classification via triplet networks. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2018, 11, 220–237.

[56] Yu Y., Liu F. Aerial Scene Classification via Multilevel Fusion Based on Deep Convolutional Neural Networks. IEEE Geosci. Remote Sens. Lett. 2018, 15, 287–291.