International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 2
ISSN: 2088-5334

Efficient Supervised Features Learning for Remote Sensing Image Classification

Sarah Qahtan Mohammed Salih[a,*], Abdul Sattar Arif Khamas[b], Ramlan Mahmod[c]

[a] *Computer Center Department, Middle Technical University, Baghdad, Iraq*
[b] *Radiology Department, Middle Technical University, Baghdad, Iraq*
[c] *Computer Science Department, University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia*
*Corresponding author: *sarah.qahtan@mtu.edu.iq*

***Abstract*— The features extracted from the fully connected (FC) layers of a convolutional neural network (ConvNet or CNN) can provide accurate classification results as long as the labelled datasets are large enough. On the other end, high accuracy remote sensing image (RSI) classification is demanded various implementations such as urban planning, environmental monitoring, and geographic image retrieval. Many studies have been presented in this domain; however, satisfactory classification accuracy is yet to be achieved. In this study, the proposed method of fine-tuning the pre-trained ConvNets (GoogleNet, VGG16, and ResNet50) on RSI, extracting features from the last fine-tuned FC layer of these networks and reprocess the extracted features for classification by SVM, produced high classification accuracy. Extensive experiments have been conducted on three RSI datasets: the NWPU, AID, and PatternNet. Comparative results over the selected datasets demonstrate that our method considerably outperforms the state-of-the-art best-stated results. Also, the overall accuracy (OA) and confusion matrix report quantitative evaluation. Our best outcomes from the first part were 99.54%, 94.60%, and 94.83% on the PatternNet, NWPU, and AID datasets, respectively, achieved by fine-tuned ResNet50. Moreover, the best classification accuracies with training ratios 20% and 50% on the AID dataset, 10% and 20% on the NWPU dataset, and the 10%, 20%, 50% and 80% on PatternNet dataset were 95.72%, 97.53%, 96.19%, 96.85%, 99.60%, 99.56%, 99.75% and 99.80% respectively. The classification performance of each class was estimated using a confusion matrix for the three datasets.**

***Keywords*— Convolutional neural networks; remote sensing; image classification; feature extraction; pre-trained; fine-tuned.**

Manuscript received 6 Mar. 2020; revised 12 Oct. 2020; accepted 25 Jan. 2021. Date of publication 30 Apr. 2021.

I. INTRODUCTION

With the rapid improvement of remote sensing technologies, large databases of high-resolution RSIs are becoming accessible. RSIs have received remarkable attention lately. They can be employed in a wide range of fields, for instance, urban planning, land surveying, computer cartography, geographic image retrieval, and others [1]–[4]. A unique label has been assigned to the images (e.g., forest or beach) that moved the research of RSI classification to more semantic understanding than pixel-level interpretation [1]. However, the RSI classification problem's challenge is the different orientations and scales of the objects; and the scene images often highly complex spatial structures with similar interclass and intraclass variability. Due to these challenges, there have been increased past years, and researchers have proposed various approaches to solve scene classification problems [5]–[16].

Recently, the massive success of feature representation of various convolutional ConvNets has been inspired by the computer vision community [17]–[22], many ConvNet models have been introduced for RSI classification [23]–[27]. The ConvNet models (e.g., ResNet [28], GoogleNet [29], VGG16 [30], and AlexNet [31]) can accomplish more excellent classification performance since pre-trained ConvNet can extract more illustrative features compared to the traditional method (e.g., the intensity of pixel-level interpretation or color histogram). The performance of the RSI classification can significantly affect the features extraction step. These features can be classified into three main classes [1]: handcrafted-feature, unsupervised-feature-learning, and deep-feature-learning methods.

The handcrafted-feature method (e.g., HOG [32], color histograms [33], and GIST [34]) focuses on various characteristics, such as texture, shape, and color information. The unsupervised-feature-learning method (e.g., K-means clustering [35], autoencoder [36], and LLE [37]) learns features automatically from unlabeled input data, and remarkable progress has been made by this methods to solve a scene classification problem. In more recent years, deep-feature-learning methods show the extraordinary capability of feature representation by automatically learning features from the raw input data [38]–[41]. One of many deep-learning architectures is the ConvNet, which has been used for various complex problems such as image classification and scene-recognition. The structure of ConvNet consists of an input and an output layer, along with a series of hidden layers, which are a convolutional (Conv), pooling (Pool), and FC layers [1]. The low-level features like corners, lines, and edges can be extracted from the first Conv layer. Conversely, complex features can be extracted from higher-level layers; then, the extracted features robust against distortion and noise in the Pool layers. In non-linear layers, a trigger function such as rectified linear units (ReLUs) is used on each hidden layer to indicate the diverse identification of likely features. FC layers are used to sum weighting of the previous layer of features to determine a precise target result [42].

Despite the accomplishment of ConvNet, there are many challenging RSI classification problems. Some of these challenges are the small scale of RSI datasets and different real-world conditions (e.g., weather, illuminations, and seasons). In addition to the difference in resolutions, object poses, viewpoints, and backgrounds. In this paper, three fine-tuned ConvNets (GoogleNet [29], ResNet50 [28], and VGG16 [30]) have been proposed to improve the performance of RSI classification, three approaches of ConvNet (GoogleNet [29], ResNet50 [28], and VGG16 [30]). The fine-tuned ConvNets have been used to transfer features; each of them is fine-tuned on three RSI datasets. Then, we extract features from the last fine-tuned FC layer of each ConvNets. The classification has been done by using a support vector machine (SVM). We evaluate our methods by comparing all the results of the fine-tuned models with state-of-the-art methods. Also, two metrics, the OA and confusion matrix, are used for quantitative evaluation.

A. Scene Classification

Generally speaking, the existing scene classification can be categorized based on features extraction into three primary levels: 1) low- level methods can extract simple features such as spectral, shape, structure, texture features; 2) mid-level methods are suitable to represent the structures of complex images; 3) high-level methods can consider the most efficient groups for extracting complex textures and structures [28]. The ConvNets (high-level feature methods) are among the most commonly used deep-learning algorithms, so in this paper, we only focus on high-level feature methods.

A pre-trained ConvNets are models that pre-trained on a large benchmark dataset as in ImageNet [43]. The researchers commonly imported and used models from published works (e.g., Nasnetlarge, Squeezenet, Densenet), as the computational cost of training these models. Penatti, Nogueira, and Dos Santos [44] evaluated the generalization ability of ConvNets models (CaffeNet [45] and OverFeat [46]), in the scenario of RSI classification. Lately, many large RSI datasets have been pre-trained and accomplished higher accuracies compared to the low-level and mid-level features methods. Examples of these datasets are PatternNet [4], NWPU [1], and AID [3].

Training the ConvNet models from scratch is another group of high-level feature methods, and it can give the most control over the network. However, it needs a considerable amount of training data to understand the variation of features, and the training times are often longer than pre-trained models. Some authors examined the performance of ConvNets training-from-scratch in the area of RSI classification [40], [47]. They found that the classification accuracy gets decent compared to the pre-trained ConvNet models, though they returned that to the limitation of training data.

Another group of researchers used the fine-tuned ConvNet models on the RSIs to elicit features for classification [1], [40], [48]. Generally, they used well-known fine-tuned ConvNet models, and they found the accuracies of the classification get much more remarkable than both pre-trained ConvNet models and training from scratch. A novel technique for "scene classification through via networks" was studied by pre-training the network on ImageNet and then fine-tuned by the selected datasets [49].

An alternative technique reprocessed the features extracted from the pre-trained ConvNet instead of directly used these features. Marmanis et al. [50] offered a two-stage classification, a set of reprocessing features were extracted from the pre-trained OverFeat and followed by transferring them along with their class labels into a supervised CNN classifier. Moreover, a linear transformation of deep features has been proposed to learn the discriminative convolution filter applied to each local patch separately [51]. Also, another study [52] obtained intensely local descriptions by extracting convolutional features from two RSI datasets by using Fisher encoding and Gaussian mixture model clustering followed by linear support vector machine (SVM) classification [53]. In another work, the deep spatial features were studied at multiple scales by extracting them from pre-trained CNNs[54]. The result of multiple scales is used for visual words encoding, correlogram extraction, correlation encoding, and classification.

B. Datasets

In the past years, different research groups have been performed scene classification by introducing several high-resolution RSI datasets to evaluate other methods in the machine-learning field. In this section, the authors will briefly review some of these datasets. The number of classes, the total number of images in the datasets, the number of images per each class, images resolution, and size of images are shown in Table 1.

Most of these datasets imported the images from Google Earth Engine, excluding the Brazilian Coffee scene [44] dataset cropped from SPOT satellite images. So far, UCM [11] has been the most popular and widely used for RSI scene classification. However, the authors selected three datasets for the experiment in this study, which are NWPU [1], AID [3], and PatternNet [4].

II. Material and Method

The methodology started by presenting the architecture of the selected ConvNets. Then, the proposed method consisted of three parts, as shown in Fig. 1. The first part fine-tunes the pre-trained ConvNets (e.g., GoogleNet [29], ResNet50 [28], and VGG16 [30]) on the RSI datasets (e.g., NWPU, AID, and PatternNet). The second part extracts features from the last fine-tuned FC layer. The last part reprocesses the second part's features for classification by support vector machine (SVM).

TABLE I
Comparisons of Some RSI Datasets

Datasets	No. of classes	Total number of images	No. of images per each class	Size of images
NWPU [1]	45	31,500	700	256 × 256
SAT_4 & SAT_6 [55]	500,000 (SAT_4) + 405,000 (SAT_6) patches			28 × 28
WHU-RS19 [56]	19	1,005	~50	600×600
AID [3]	30	10,000	200 - 400	600 × 600
UCM [11]	21	2,100	100	256 x 256
RSI-CB256 and RSI-CB128 [57]	35 and 45	~24,000 and 36,000	~690 and 800	256 × 256 and 128 × 128
RSSCN7 [58]	7	2,800	400	400 × 400
RSC11 [59]	11	1,232	~100	512 × 512
Brazilian Coffee [44]	2	2,876	1,438	64 × 64
SIRI-WHU [10]	12	2,400	200	200 x 200
PatternNet [4]	38	30,400	800	256 x 256

A. Convolution Neural Network Architecture

GoogleNet [29] is one of the well-known ConvNets presented by Szegedy et al. that took first place due to a 6.67% error rate in ILSVRC2014. It consists of 9 inception modules, where each module has 6 Convs layers and 1 Pool layer concatenating the outputs to achieve a multi-scale features extraction in each module. Each inception module runs simultaneously four (1 x 1 Conv), one (3 x 3 Conv), one (5 x 5 Conv) with one (3 x 3 pool). The last FC layer contains 1024 neurons.

VGG16 [30] is another representative ConvNet architecture proposed by Simonyan & Zisserman that won ILSVRC2014. VGG accomplishes the top 5 with 92.7% accuracy on ImageNet [60]. It has 21 layers, thirteen Conv, five Pool, and three FC but only sixteen weight layers, and the first input layer is fixed size pixels 224 × 224 with 3 color channels (red, green, and blue). The filters number of Conv1, Conv2, and Conv3 are 64, 128, and 256, respectively, whereas Conv4 and Conv5 have 512 filters. The filter size in all Conv layers is 3 × 3 pixels, and the stride of the Conv process is fixed to 1 pixel. Pool layers follow some of the Conv layers; these layers are 2 × 2 pixels with stride 2. The first and second FC contains 4096 neurons, while the last one contains 1000 neurons.

ResNet-50 [28] introduced by He *et al*., achieved the best performance on ImageNet in ILSVRC-2015. It has four blocks with 3, 4, 6, and 3 such units, respectively. The image input size is 224 × 224 pixels with three color channels, followed by Conv and Pool layers with 7 × 7 and 3 × 3 pixels, respectively. There are three layers in all the three units in block one, where each unit consists of 1 x 1 Conv, 3 x 3 Conv, and 1 x 1 Conv with filter size 64, 64, and 128, respectively. The width of Convs in the final block is doubled, and the size of the input is reduced by half. As a final point in the network, an average pooling layer followed by the FC layer contains 2048 neurons.

B. Fine-Tuning of Pre-trained ConvNets

In the first part, the pre-trained network loaded by using the deep-learning toolbox model for GoogleNet, ResNet50, and VGG16 networks. We explored the network's architecture and some details about the network layers by using the "Analyze Network" Matlab-Function. We found that the first layer in each one of the used networks is the image input layer. The size of the images is 224 × 224 for the three networks; therefore, the selected dataset images are rescaled. For each experiment, the datasets split into 70% for the training phase, 30% for the testing phase, as shown in Table 2.

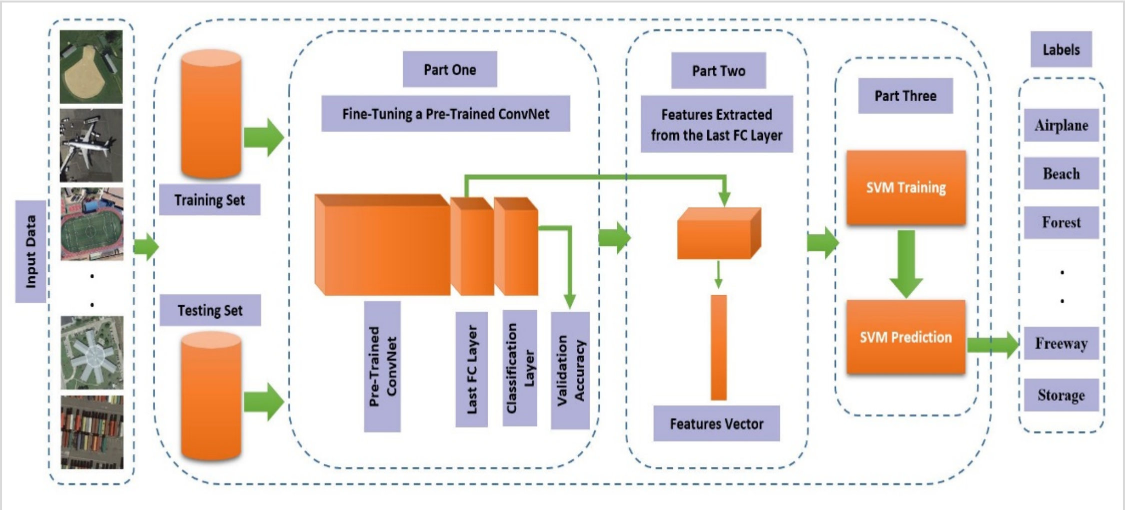


Fig. 1 Flow chart of the proposed method that consists of three parts: 1) fine-tunes the pre-trained ConvNets on the RSI datasets; 2) features extraction; 3) SVM classification.

TABLE III
DATASET SETTING FOR EXPERIMENTAL RESULTS

DataSet	Total Images	Training (70%)	Testing (30%)	No. of iterations per epoch	Maximum iterations
PatternNet [4]	30,400	21,280	9,120	425	4250
NWPU [1]	31,500	22,050	9,450	441	4410
AID [3]	10,000	7,000	3,000	140	1400

The last learnable layers (the fully-connected layer) loss3-classifier, fc1000, and fc8 in GoogleNet, ResNet50, and VGG16 networks respectively replaced with new layers (fine-tuned) to retrain the pre-trained networks. The new fully-connected layer trained weights are 45, 30, and 38 according to the number of classes in the selected datasets. The final classification layer in the three networks replaced with new layers to fit our datasets (output size that equals the number of classes in each dataset). After that, to minimize the time consuming of the network's training, the earlier transferred layers (first ten weighted layers) were set to zero by freezing the weights of these layers—finally, the fine-tuned network used to validate the classification and to calculate the classification accuracy. We used ten epochs in each experiment, the learning rate and the mini-batch size set 0.001 and 50, respectively. The number of iterations (*I*) per each epoch and the maximum iterations (*Max Iter*) is calculated using equations 1 and 2.

$$I = no.of\ trained\ image/the\ mini - batch\ size \qquad (1)$$

$$Max\ Iter = I * no.of\ epoch \qquad (2)$$

The second part of our method starts by loading and analyzing our fine-tuned networks. For features extraction, some researchers(e.g.,[40] and [44]) extracted an activation vector from the last FC layer. Thus, in our experiment, we extracted features from loss3-classifier, fc1000, and fc8 in GoogleNet, ResNet50, and VGG16 networks, respectively. The last part of our method, for the three selected datasets SVM using fitcecoc (Statistics and Machine Learning Toolbox), is used for training and classification. The training and classification were performed to the fine-tuned networks using the second part of our method's features.

The training and testing samples are selected randomly for each dataset. The accuracies are reported by repeating each experiment ten times and calculating the average.

C. Experiment Setup and Analysis

Our experiments are conducted on a Lenovo Y50 laptop with Windows 10, 64bit, and MATLAB R2019a, Intel Core i7-4710HQ processor with a graphics processing unit (GPU) NVIDIA GeForce GTX, CPU (2.50 GHz), and 16 GB RAM. We evaluated our proposed method on three public RSI datasets, which were: NWPU [1], AID [3], and PatternNet [4]. Fig. 2, 3, and 4 show sample images from these datasets along with their classes.

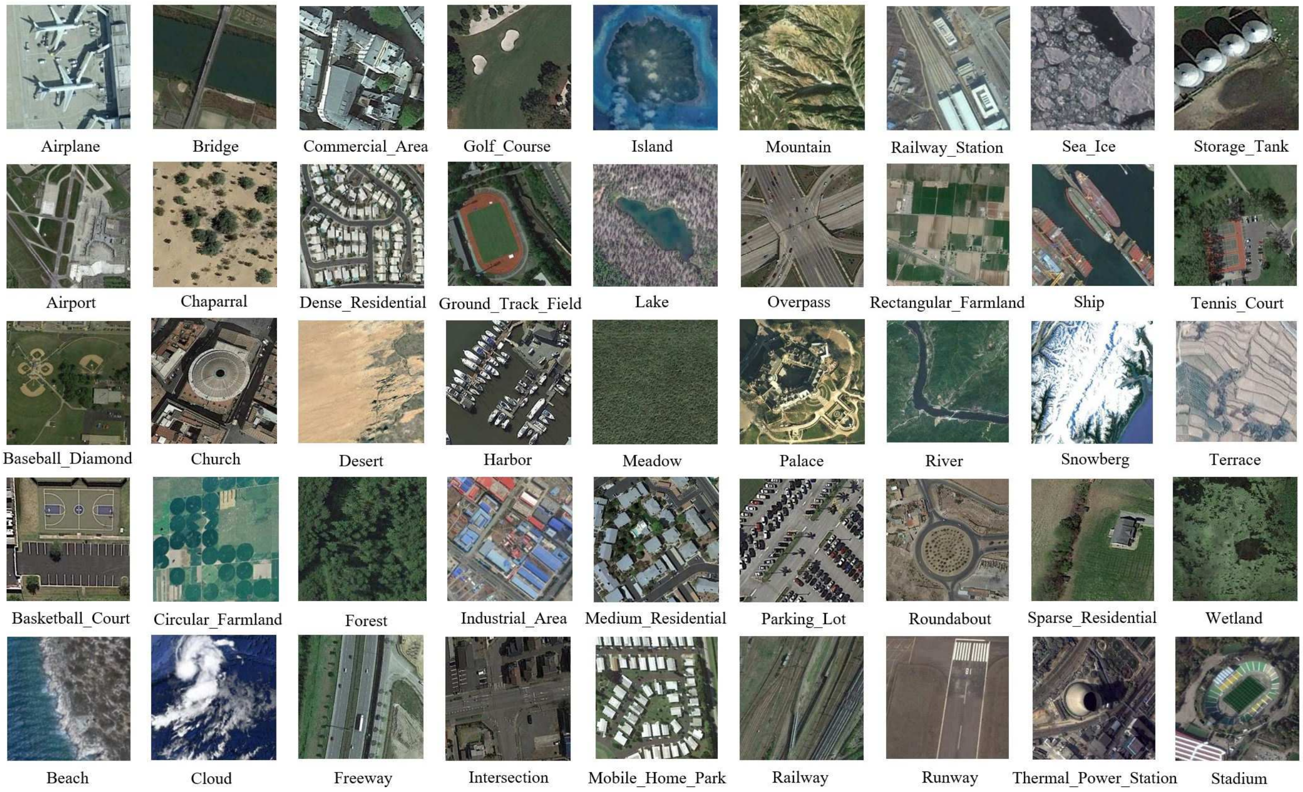


Fig. 2 Some images from NWPU dataset

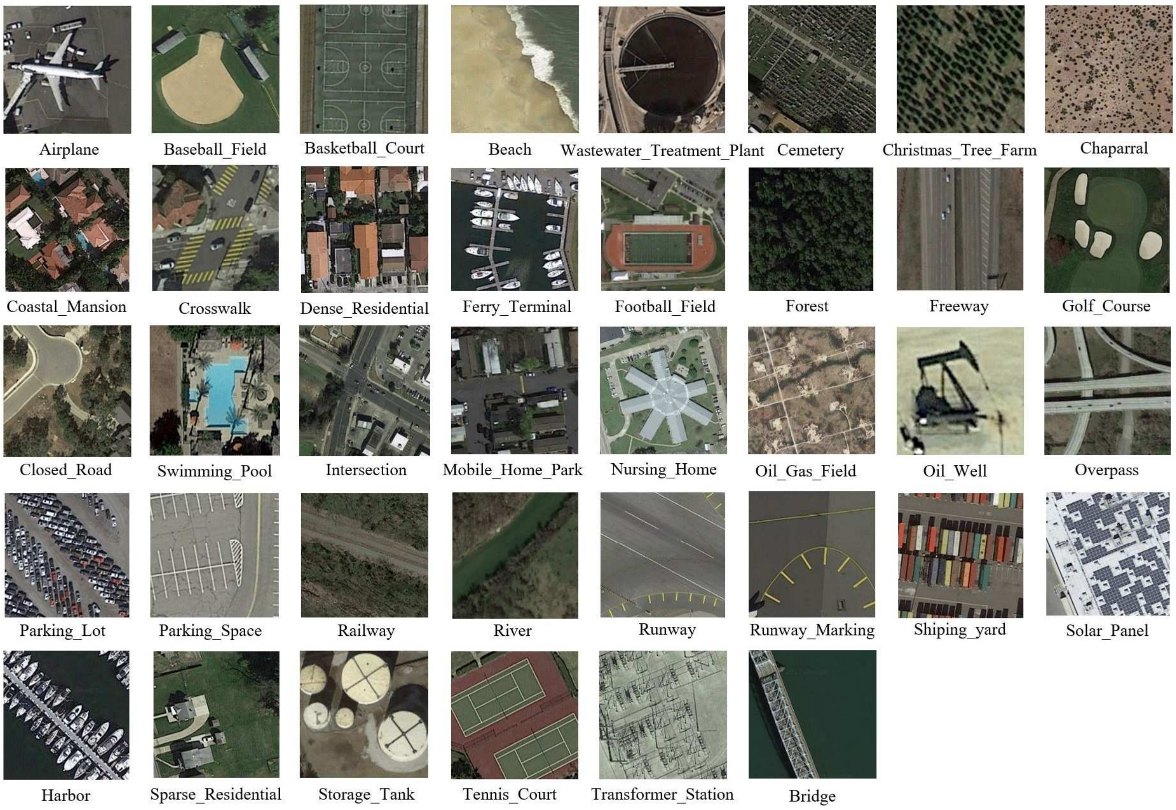


Fig. 3 Some images from PatternNet dataset

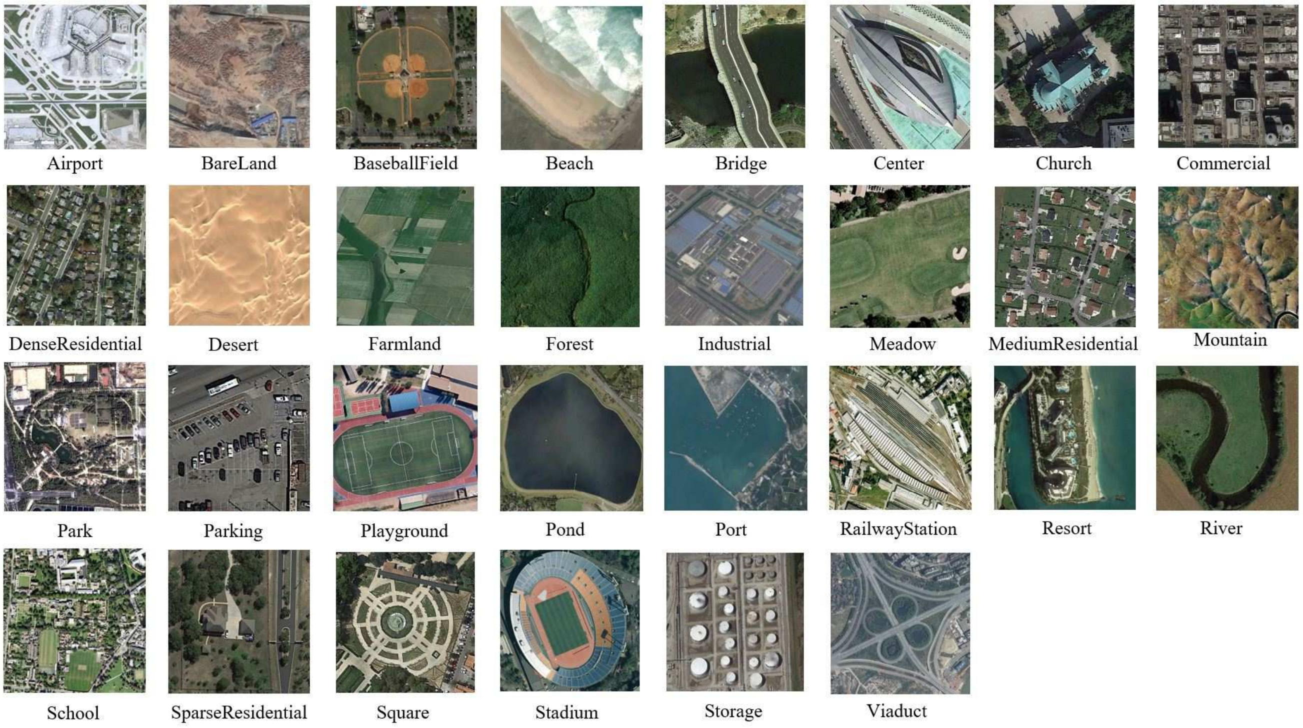


Fig. 4 Some images from AID dataset

The details of each dataset presented in Table 1. First, to evaluate our proposed method, we compared the validation accuracy of our fine-tuned ConvNets in the first part of our method with previous methods. The experiment is set by following the description in [22]; three training-and-testing ratios are used in this experiment. Thus, the AID dataset's training ratios are 10% and 50%; the NWPU dataset is 10% and 20%. While the training ratios on PatternNet are 10%, 20%, 50%, and 80%. The reason for that, the authors could not find any previous work split PatternNet dataset based on any of these ratios. Therefore, we selected all the training-and-testing ratios used in our experiment plus 80% vs. 20% [22]. For quantitative evaluation, we used two widely selected metrics which are OA and confusion matrix. The experiments are repeated ten times, and the final performance is measured for each training-and-testing ratio. Equations (3) and (4) show the OA calculation, where *M* is the number of the correct prediction in the test set, and *N* is the total number of classes in the test set. Furthermore, we examine the confusion matrix of the three datasets to estimate each class's classification performance where the rows represent the actual classes, while the columns represent the predicted classes.

$$Mean = \frac{1}{10}\,\frac{\sum_{n=1}^{10} M}{N} \tag{3}$$

$$std = (\frac{1}{10-1}\sum_{n=1}^{10}(M - Mean)^2)^{1/2} \tag{4}$$

III. Results and Discussion

A. Comparisons with Other Fine-tuned ConvNet Method

To illustrate our proposed method's effectiveness, we compared our results (the validation accuracy of the fine-tuned networks) with previous methods that have reported the classification accuracy on the selected datasets. As shown in Table 3, the accuracy values [13], [25] on the AID dataset reached 94.42% and 94.3%, respectively, which achieved the closest accuracy to our accuracy (94.83%) by using fine-tuned ResNet50. The study presented by He *et al.* [13] suggested a mechanism for the classification and the feature learning, where their input is a group of HRRS (High-resolution remote sensing) images. Then they processed these images by a series of spatial-scale-aware blocks to obtain high-level feature vectors for the classification. They used two datasets the UCM, and the AID, and their work did not test on NWPU and PatternNet datasets. A multilayer-stacked-covariance-pooling (MSCP) proposed by [25] conducted on three RSI datasets (e.g., UC Merced Land Use, AID, and NWPU). They extracted multilayer features obtained by AlexNet and VGG16, stacked them together, and found a covariance matrix for the stacked features. Our fine-tuned GoogleNet and VGG16 on the AID dataset could achieve better results compared to the study presented by Xia *et al.* [3]. A multi-branch-neural-network (MB-Net) consists of four datasets [14]: UC Merced Land Use, AID, PatternNet, and NWPU. They learned invariant feature demonstrations from several labeled images with one unlabeled image. In particular, our fine-tuned ResNet50 method boosts the accuracy overall previous methods except ResNet50 [15], which achieved the highest accuracy of 95.7% on NWPU.

In comparison, the best accuracy we could achieve on NWPU dataset was 94.60% by using fine-tuned ResNet50. As can be seen, in general, our method achieved the best performance, fine-tuned ResNet50 achieve competitive results, even though Fine-Tuned GoogleNet and Fine-Tuned VGG16 show the worst results on AID. It is interesting to note that the performance on PatternNet is expressively better than that on AID and NWPU.

TABLE IIII
Comparisons to the Previous Methods

Models	Validation Accuracy		
	PatternNet	**NWPU**	**AID**
HRRS learning strategy [13]	-	-	94.3%
MSCP [25]	-	88.93%	94.42 %
GoogleNet [3]	-	-	86.39%
VGG16 [16]	-	88.62%	-
ResNet50 [15]	-	95.7%	-
MB-Net [14]	98.05	76.38%	91.46%
The proposed method with Fine-Tuned GoogleNet	96.39%	91.66%	88.40%
The proposed method with Fine-Tuned VGG16	98.83%	91.10%	89.63%
The proposed method with Fine-Tuned ResNet50	**99.54%**	**94.60%**	**94.83%**

B. Comparisons with Other Classification Methods

Further comparisons have been made based on three different training-and-testing ratios on the selected datasets. The training-and-testing ratios on AID are 20% and 50% for training, 80% and 50% for testing. On NWPU are 10% and 20% for training, 90% and 80% for testing, respectively. The training-and-testing ratios on PatternNet are 10%, 20%, 50%, and 80% for training; 90%, 80%, 50% and 20% for testing respectively. Table 4 reported the mean accuracy and standard deviation for each training-and-testing ratio over ten times. Also, comparisons of different techniques using similar training-and-testing ratios are also reported. Saliency Two-Stream Network (SAL-TS-Net) was derived from fine-tuned pre-trained GoogleNet extracted features from the last Pool layer and used Extreme Learning Machine (ELM) classifier. SAL-TS-Net computes the means and standard deviations under different training-and-testing ratios.

TABLE IVV
Comparisons of Classification Results on AID and NWPU Datasets Achieved in this Study with Very Resent RSI Classification Methods [22, 25]

Model	**OA (mean ± standard deviation)**			
	AID		**NWPU**	
	20%	**50%**	**10%**	**20%**
SAL-TS-Net [22]	94.09±0.34	95.99±0.35	85.02±0.25	87.01±0.19
VGG16+MSCP+MRA [25]	92.21±0.17	96.56±0.18	88.07±0.17	90.81±0.13
GoogleNet + SVM	92.02±0.97	94.53±0.97	93.27±0.39	94.55±0.47
VGG16 + SVM	95.21±0.91	97.19±0.72	**96.19±0.49**	**96.85±0.72**
ResNet50 + SVM	**95.72±0.99**	**97.53±0.80**	95.61±0.57	96.59±0.38

VGG16+MSCP+MRA [25] is used a pre-trained VGG16 to extract multilayer feature plus "multiresolution analysis to improve classification accuracy" (MRA) further. The extracted features are stacked together and classified by

SVM. As shown in Table 4, the methods by previous studies [22], [25] are slightly better than our approach using GoogleNet+SVM on AID. However, our methods (VGG16+SVM and ResNet50+SVM) showed better classification or competitive performance on the same dataset, and the best results are marked in bold. All the proposed methods on NWPU outperform the results by previous studies [22], [25].

TABLE V
COMPARISONS OF CLASSIFICATION RESULTS BASED ON DIFFERENT TRAINING-AND-TESTING RATIOS ON PATTERNNET DATASET

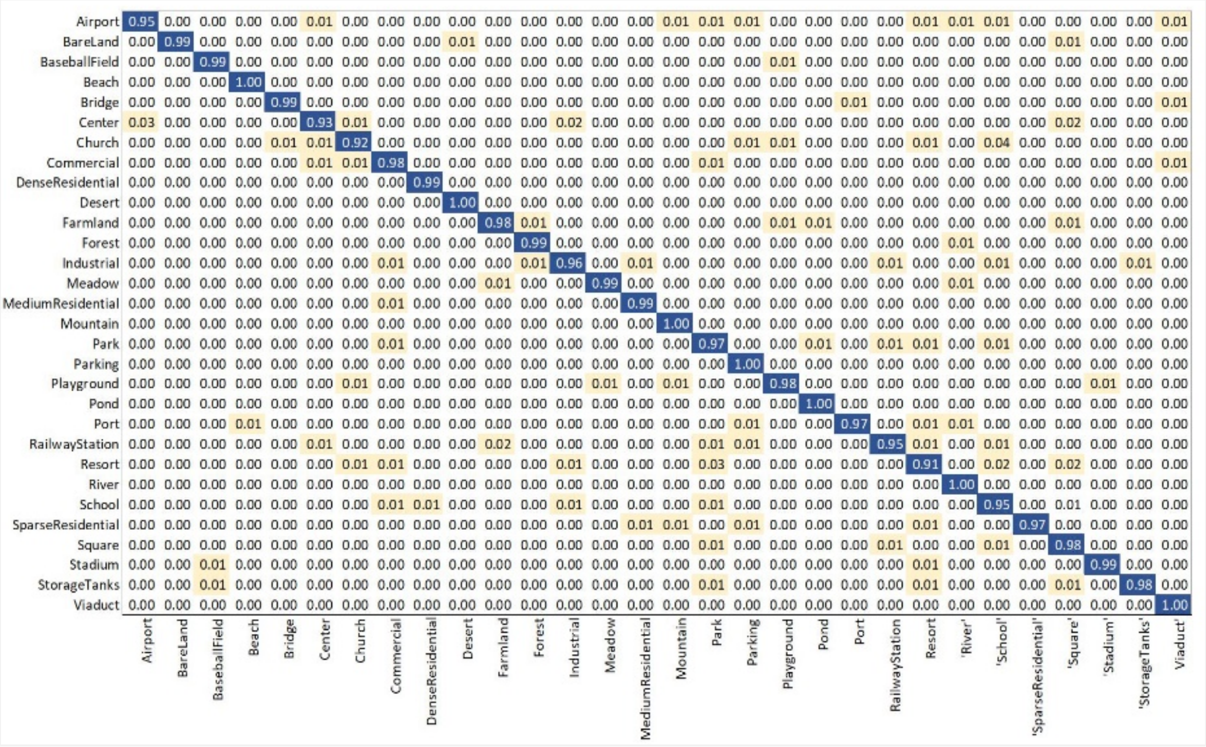
Model	OA (mean ± standard deviation)			
	PatternNet			
	10%	20%	50%	80%
GoogleNet + SVM	98.53±0.70	99.02±0.74	99.08±0.57	99.54±0.30
VGG16 + SVM	**99.60±0.26**	99.52±0.38	99.57±0.34	99.73±0.18
ResNet50 + SVM	99.52±0.30	**99.56±0.44**	**99.75±0.15**	**99.80±0.14**

In Table 5, different training-and-testing ratios on the PatternNet dataset are summarized; however, we could not find any previous method to compare. Yu and Liu [22] used 50% and 80% training ratios on UCM dataset, and we followed the split techniques [22] for the PatternNet dataset. Considering the PatternNet dataset compared to the UCM dataset, we added additional training ratios (10% and 20%) on PatternNet dataset. All the proposed method that applied to PatternNet dataset present a very high accuracy of 98.53-99.80.

C. Confusion Analysis

In this section, the confusion matrix and the classification accuracy are reported for each class in the selected datasets. Considering the limitation of the article space, we only show the confusion matrixes for the AID dataset 50% training ratio using the ResNet50+SVM, for NWPU dataset, 20% training ratio using the VGG16 + SVM, which are the best method achieved on these datasets. However, for the PatternNet dataset, 10% training ratio using GoogleNet+SVM is selected, which is the lowest result we got in the chosen dataset.

There are 27 among 30 AID classes that have classification accuracies higher than 0.95; the classification accuracies of the beach, desert, mountain, parking, pond, river, and viaduct can reach 1, as shown in Fig. 5



	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	0.95	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
BareLand	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
BaseballField	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Beach	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bridge	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Center	0.03	0.00	0.00	0.00	0.00	0.93	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
Church	0.00	0.00	0.00	0.00	0.01	0.01	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.04	0.00	0.00	0.00	0.00	0.00
Commercial	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
DenseResidential	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Desert	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Farmland	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
Forest	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Industrial	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.96	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
Meadow	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
MediumResidential	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Mountain	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Park	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Parking	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Playground	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
Pond	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Port	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.97	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
RailwayStation	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.95	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Resort	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.02	0.00	0.02	0.00	0.00	0.00
River	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
School	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.01	0.00	0.00	0.00
SparseResidential	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.97	0.00	0.00	0.00	0.00
Square	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.98	0.00	0.00	0.00
Stadium	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.99	0.00	0.00
StorageTanks	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.98	0.00
Viaduct	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Fig. 5 Confusion matrixes for AID dataset with 50% training ratio by using the ResNet50+SVM

Also, bare land, baseball field, bridge, dense residential, forest, meadow, medium residential, and the stadium can reach 0.99. This method can acquire good performance in some categories that are challenging to distinguish. For example, from the results gained by using MSCP+VGG16 [25], they got low accuracies under some scene-categories, such as center (0.85), resort (0.82), school (0.85), square (0.89), industrial (0.89), and park (0.86). Our classification accuracy for the same scene-categories are boosted, where center (0.93), resort (0.91), school (0.95), square (0.98), industrial (0.96) and park (0.97).

For the NWPU dataset, 36 among 45 classes can achieve the classification accuracy exceeding 0.95; precisely, the beach, chaparral, circular farmland, forest, harbor, snowberg, storage tank, and thermal power station can get classification accuracies 0.99-1. The confusion matrix for VGG16 + SVM is present in Fig. 6 compared with the one present using texture-coded two-stream with the fusion model [22] under the same training-and-training ratio.

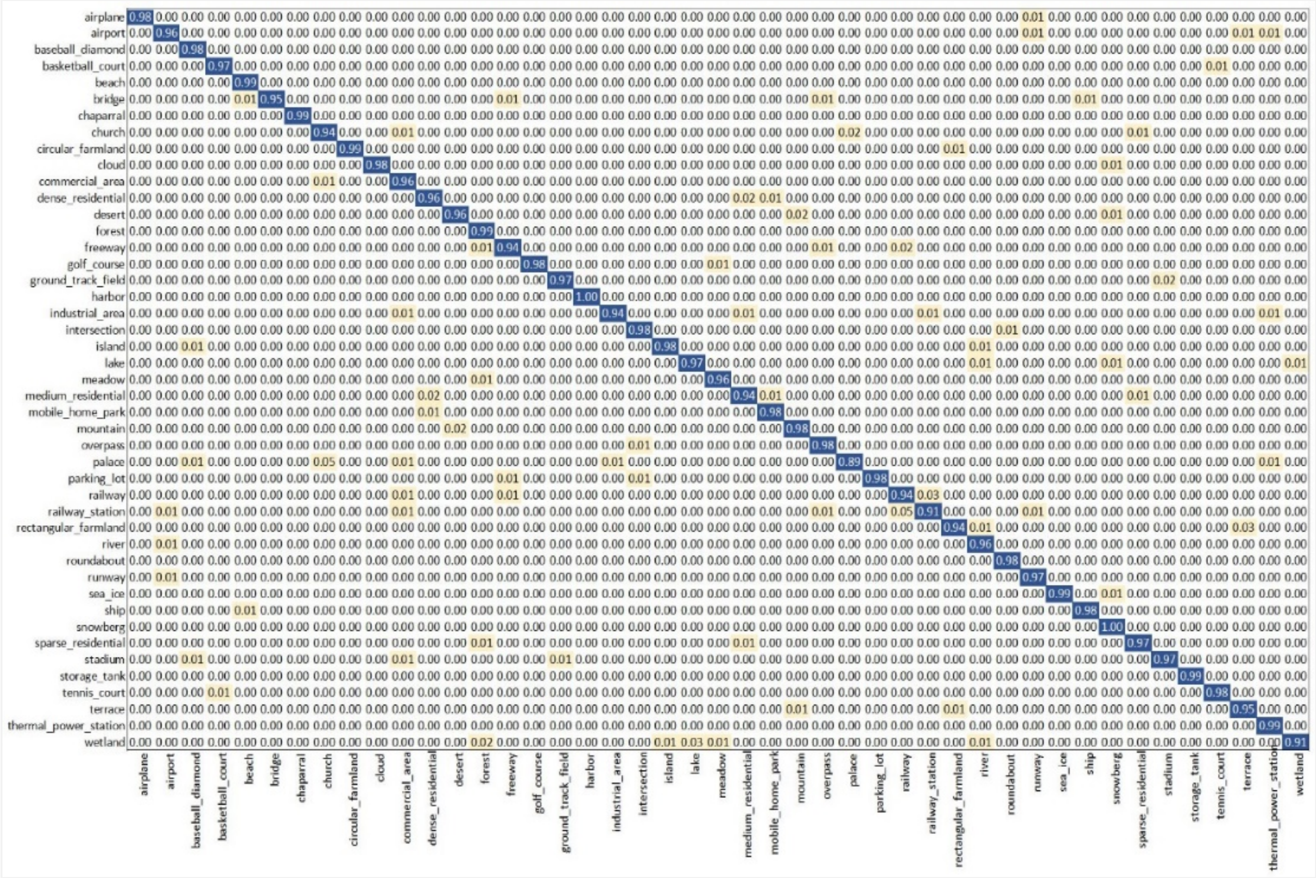


Fig. 6 Confusion matrixes for NWPU dataset with 20% training ratio by using the VGG16+SVM

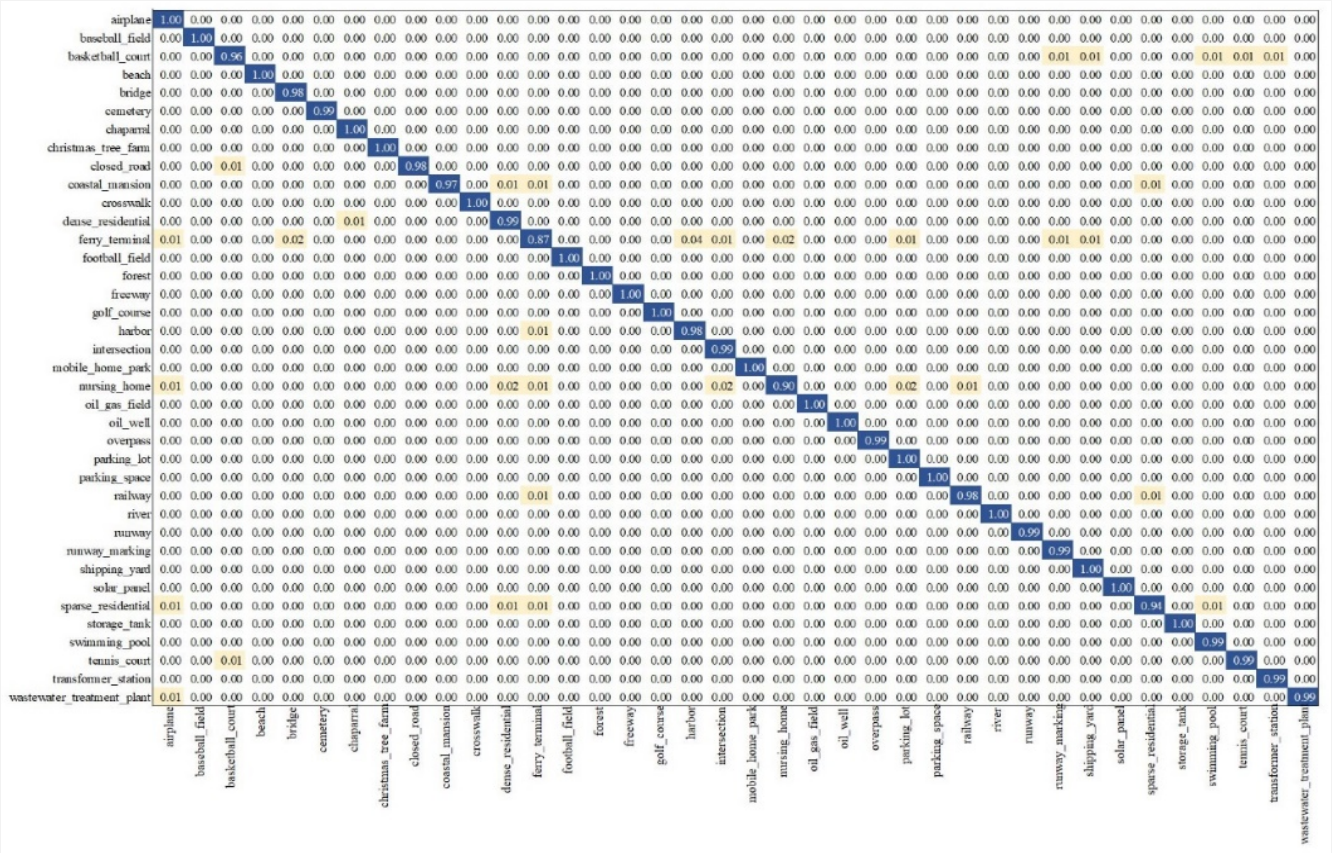


Fig. 7 Confusion matrixes for PatternNet dataset with 10% training ratio by using the ResNet50+SVM

For instance, the low accuracy under baseball diamond, basketball court, church, commercial area, dense residential, freeway, industrial area, medium residential, palace, railway station, river, tennis court and wetland scene-categories were 0.81, 0.73, 0.64, 0.76, 0.80, 0.73, 0.75, 0.77, 0.61, 0.78, 0.80, 0.72 and 0.77 respectively by [22] and 0.98,0.97, 0.94, 0.96, 0.96, 0.94, 0.94, 0.94, 0.89, 0.91, 0.96, 0.98 and 0.91 respectively by the proposed method.

Fig. 7 presents the confusion matrix for the PatternNet dataset using the ResNet50+SVM under the training ratio of 10%. 19 among 38 scene-categories can accomplish the classification accuracy equal 1; another 14 scene-categories among 38 can achieve the classification accuracy between 0.98-0.99. The lowest accuracies we got by using GoogleNet+SVM on ferry terminal (0.87), nursing home (0.90), sparse residential (0.94), and basketball court (0.96).

IV. Conclusion

This paper proposed supervised feature learning based on pre-trained ConvNet (e.g., GoogleNet, VGG16, and ResNet50) for RSI classification. The proposed method was first fine-tuning the pre-trained ConvNets on three publicly available RSI datasets: NWPU, AID, and PatternNet. Then, we extracted the features from the last fine-tuned FC layer. Finally, we reprocessed the extracted features for classification by using SVM. Comprehensive experiments and comparisons conducted with previous approaches confirm the efficiency of the proposed methods. Our best result, by fine-tuned ResNet50, can achieve 99.54%, 94.60%, and 94.83% on the PatternNet, NWPU, and AID datasets, respectively. Furthermore, our best classification accuracies were 95.72%, 97.53%, 96.19%, 96.85%, 99.60%, 99.56%, 99.75% and 99.80% with training ratios 20% and 50% on the AID dataset, 10% and 20% on the NWPU dataset, and the 10%, 20%, 50% and 80% on PatternNet dataset, respectively. Then, we used a confusion matrix to estimate each class's classification performance for the three datasets. According to the confusion matrix, the classification accuracy of 15, 8, 28 classes from AID, NWPU and PatternNet dataset could reach more than 0.99. Besides, we raise some challenging classes from AID and NWPU datasets compared to the same classes proposed by [22] and [25].

On the other hand, we could not find any previous work reports on the confusion matrix of the PatternNet dataset; therefore, we just report our confusion matrix. In the future, we are planning to extend our method by testing other ConvNets and classification methods on RSI. Also, we are planning to merge the same classes of the RSI datasets to explore the impact on learning and classification.

References

[1] G. Cheng, J. Han, and X. Lu, "*Remote sensing image scene classification: Benchmark and state of the art*," Proceedings of the IEEE, vol. 105, pp. 1865-1883, 2017.

[2] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, "*Extinction profiles fusion for hyperspectral images classification*," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 1803-1815, 2017.

[3] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, et al., "*AID: A benchmark data set for performance evaluation of aerial scene classification*," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 3965-3981, 2017.

[4] W. Zhou, S. Newsam, C. Li, and Z. Shao, "*PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval*," ISPRS journal of photogrammetry and remote sensing, vol. 145, pp. 197-209, 2018.

[5] X. Bian, C. Chen, L. Tian, and Q. Du, "*Fusing local and global features for high-resolution scene classification*," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, pp. 2889-2901, 2017.

[6] G. Cheng, J. Han, L. Guo, and T. Liu, "*Learning coarse-to-fine sparselets for efficient object detection and scene classification*," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1173-1181.

[7] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "*Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images*," IEEE Transactions on Geoscience and Remote Sensing, vol. 53, pp. 4238-4249, 2015.

[8] 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, p. 483, 2016.

[9] X. Lu, X. Zheng, and Y. Yuan, "*Remote sensing scene classification by unsupervised representation learning*," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 5148-5157, 2017.

[10] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "*Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery*," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, pp. 2108-2123, 2015.

[11] Q. Zhu, Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang, "*Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery*," IEEE Geoscience and Remote Sensing Letters, vol. 13, pp. 747-751, 2016.

[12] J. Zou, W. Li, C. Chen, and Q. Du, "*Scene classification using local and global features with collaborative representation fusion*," Information Sciences, vol. 348, pp. 209-226, 2016.

[13] Z. Chen, Y. Wang, W. Han, R. Feng, and J. Chen, "*An Improved Pretraining Strategy-Based Scene Classification With Deep Learning*," IEEE Geoscience and Remote Sensing Letters, 2019.

[14] M. M. Al Rahhal, Y. Bazi, T. Abdullah, M. L. Mekhalfi, H. AlHichri, and M. Zuair, "*Learning a multi-branch neural network from multiple sources for knowledge adaptation in remote sensing imagery*," Remote Sensing, vol. 10, p. 1890, 2018.

[15] O. Sen and H. Y. Keles, "*Scene Recognition with Deep Learning Methods Using Aerial Images,*" in 2019 27th Signal Processing and Communications Applications Conference (SIU), 2019, pp. 1-4.

[16] Y. Yao, H. Zhao, D. Huang, and Q. Tan, " *Remote Sensing Scene Classification Using Multiple Pyramid Pooling*," International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 2019.

[17] Y. Feng, Y. Yuan, and X. Lu, "*Learning deep event models for crowd anomaly detection*," Neurocomputing, vol. 219, pp. 548-556, 2017.

[18] X. Lu, B. Wang, X. Zheng, and X. Li, "*Exploring models and data for remote sensing image caption generation*," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 2183-2195, 2017.

[19] W. Zhang, X. Lu, and X. Li, "*A coarse-to-fine semi-supervised change detection for multispectral images*," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 3587-3599, 2018.

[20] J. Zhu, L. Fang, and P. Ghamisi, "*Deformable convolutional neural networks for hyperspectral image classification*," IEEE Geoscience and Remote Sensing Letters, vol. 15, pp. 1254-1258, 2018.

[21] M. A. Kadhim and M. H. Abed, "*Convolutional Neural Network for Satellite Image Classification*," in Asian Conference on Intelligent Information and Database Systems, 2019, pp. 165-178.

[22] Y. Yu and F. Liu, "*Dense connectivity based two-stream deep feature fusion framework for aerial scene classification*," Remote Sensing, vol. 10, p. 1158, 2018.

[23] 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, pp. 14680-14707, 2015.

[24] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "*When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs*," IEEE transactions on geoscience and remote sensing, vol. 56, pp. 2811-2821, 2018.

[25] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "*Remote sensing scene classification using multilayer stacked covariance pooling*," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 6899-6910, 2018.

[26] J. Xie, N. He, L. Fang, and A. Plaza, "*Scale-free convolutional neural network for remote sensing scene classification*," IEEE Transactions on Geoscience and Remote Sensing, vol. 57, pp. 6916-6928, 2019.

[27] J. Zhang, C. Lu, X. Li, H.-J. Kim, and J. Wang, "*A full convolutional network based on DenseNet for remote sensing scene classification,*" Math. Biosci. Eng, vol. 16, pp. 3345-3367, 2019.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "*Deep residual learning for image recognition*," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "*Going deeper with convolutions*," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

[30] K. Simonyan and A. Zisserman, "*Very deep convolutional networks for large-scale image recognition*," arXiv preprint arXiv:1409.1556, 2014.

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

[32] N. Dalal and B. Triggs, "*Histograms of oriented gradients for human detection*," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), 2005, pp. 886-893.

[33] M. J. Swain and D. H. Ballard, "*Color indexing*," International journal of computer vision, vol. 7, pp. 11-32, 1991.

[34] A. Oliva and A. Torralba, "*Modeling the shape of the scene: A holistic representation of the spatial envelope*," International journal of computer vision, vol. 42, pp. 145-175, 2001.

[35] A. Coates and A. Y. Ng, "*Learning feature representations with k-means*," in Neural networks: Tricks of the trade, ed: Springer, 2012, pp. 561-580.

[36] G. E. Hinton and R. R. Salakhutdinov, "*Reducing the dimensionality of data with neural networks*," science, vol. 313, pp. 504-507, 2006.

[37] L. K. Saul and S. T. Roweis, "*An introduction to locally linear embedding*," unpublished. Available at: http://www. cs. toronto. edu/~ roweis/lle/publications. html, 2000.

[38] F. Özyurt, "*Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures*," The Journal of Supercomputing, pp. 1-19, 2019.

[39] R. Zhu, L. Yan, N. Mo, and Y. Liu, "*Attention-Based Deep Feature Fusion for the Scene Classification of High-Resolution Remote Sensing Images*," Remote Sensing, vol. 11, p. 1996, 2019.

[40] K. Nogueira, O. A. Penatti, and J. A. Dos Santos, "*Towards better exploiting convolutional neural networks for remote sensing scene classification*," Pattern Recognition, vol. 61, pp. 539-556, 2017.

[41] S. Chaib, H. Liu, Y. Gu, and H. Yao, "*Deep feature fusion for VHR remote sensing scene classification*," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 4775-4784, 2017.

[42] S. Hijazi, R. Kumar, and C. Rowen, "*Using convolutional neural networks for image recognition*," Cadence Design Systems Inc.: San Jose, CA, USA, pp. 1-12, 2015.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "*Imagenet large scale visual recognition challenge*," International journal of computer vision, vol. 115, pp. 211-252, 2015.

[44] O. A. Penatti, K. Nogueira, and J. A. Dos Santos, "*Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?*," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 44-51.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., "*Caffe: Convolutional architecture for fast feature embedding*," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675-678.

[46] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "*Overfeat: Integrated recognition, localization and detection using convolutional networks*," arXiv preprint arXiv:1312.6229, 2013.

[47] 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, vol. 10, pp. 745-752, 2016.

[48] 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.

[49] Y. Liu and C. Huang, "*Scene classification via triplet networks*," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, pp. 220-237, 2017.

[50] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "*Deep learning earth observation classification using ImageNet pretrained networks*," IEEE Geoscience and Remote Sensing Letters, vol. 13, pp. 105-109, 2015.

[51] N. Liu, X. Lu, L. Wan, H. Huo, and T. Fang, "*Improving the separability of deep features with discriminative convolution filters for RSI classification*," ISPRS International Journal of Geo-Information, vol. 7, p. 95, 2018.

[52] N. Liu, L. Wan, Y. Zhang, T. Zhou, H. Huo, and T. Fang, "*Exploiting convolutional neural networks with deeply local description for remote sensing image classification*," IEEE access, vol. 6, pp. 11215-11228, 2018.

[53] C.-C. Chang and C.-J. Lin, "*LIBSVM: A library for support vector machines*," ACM transactions on intelligent systems and technology (TIST), vol. 2, pp. 1-27, 2011.

[54] K. Qi, C. Yang, Q. Guan, H. Wu, and J. Gong, "*A multi-scale deeply described correlatons-based model for land-use scene classification*," Remote Sensing, vol. 9, p. 917, 2017.

[55] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "*Deepsat: a learning framework for satellite imagery*," in Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, 2015, pp. 1-10.

[56] 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, pp. 2395-2412, 2012.

[57] H. Li, C. Tao, Z. Wu, J. Chen, J. Gong, and M. Deng, "*Rsi-cb: A large scale remote sensing image classification benchmark via crowdsource data*," arXiv preprint arXiv:1705.10450, 2017.

[58] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "*Deep learning based feature selection for remote sensing scene classification*," IEEE Geoscience and Remote Sensing Letters, vol. 12, pp. 2321-2325, 2015.

[59] L. Zhao, P. Tang, and L. Huo, "*Feature significance-based multibag-of-visual-words model for remote sensing image scene classification*," Journal of Applied Remote Sensing, vol. 10, p. 035004, 2016.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "*Imagenet: A large-scale hierarchical image database*," in 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248-255.