Vol. 00, No. 00, August 2016, 1–14

Deep Learning in Remote Sensing Scene Classification: A Data Augmentation Enhanced CNN Framework

Xingrui Yu[a], Xiaomin Wu[a], Chunbo Luo[b], and Peng Ren[a]

[a] *College of Information and Control Engineering, China University of Petroleum (East China),*
66 Changjiang West Road, Qingdao, 266580, China;
[b] *Department of Computer Science, College of Engineering, Mathematics and Physical Sciences,*
University of Exeter, Exeter, EX4 4QF, United Kingdom.

(*Received 00 Month 201x; final version received 00 Month 201x*)

The recent emergence of deep learning for characterizing complex patterns in remote sensing imagery reveals its high potential to address some classic challenges in this domain, e.g. scene classification. Typical deep learning models require extremely large datasets with rich contents to train a multi-layer structure in order to capture the essential features of scenes. Compared with the benchmark datasets used in popular deep learning frameworks, however, the volumes of available remote sensing datasets are particularly limited, which have restricted deep learning methods from achieving full performance gains. In order to address this fundamental problem, this paper introduces a methodology to not only enhance the volume and completeness of training data for any remote sensing datasets, but also exploit the enhanced datasets to train a deep convolutional neural network (CNN) that achieves state-of-the-art scene classification performance. Specifically, we propose to enhance any original dataset by applying three operations: flip, translation and rotation to generate augmented data, and use the augmented dataset to train and obtain a more descriptive deep model. The proposed methodology is validated in three recently released remote sensing datasets, and confirmed as an effective technique that significantly contributes to potentially revolutionary changes in remote sensing scene classification, empowered by deep learning.

Keywords: Deep Learning; Remote Sensing Scene Classification; Convolutional Neural Network (CNN); Big Data; Data Augmentation.

1. Introduction

1.1. *Background*

Scene image analysis has been an important topic in the research literature of remote sensing. Lots of efforts have been made in the low-level analysis such as scene image

ISSN: 1548-1603 print/ISSN 1943-7226 online
© 2016
DOI: xx.xxxx/xxxxxxxxxxxxxxxxxxx

pixel fusion (Zhou and Gao 2014) and the middle-level analysis such as scene feature extraction and estimation (Tomas *et al.* 2016). Recently, as the volume of accessible remote sensing image data increases tremendously, the high-level scene image analysis such as scene classification has attracted especial research interest. The study of scene classification aims at developing intelligent algorithms that categorize individual remote sensing images into different scene classes. One widely accepted way to develop such classification algorithms is to train a certain classification model based on training data, i.e. remote sensing images with known scene class labels. The trained model is then used for predicting the scene class labels of unknown remote sensing images. In this regard, various machine learning models, e.g. spectral mixture analysis (Tang and Pannell 2009) and sparse coding (Cui *et al.* 2015), have been exploited for classifying remote sensing images.

Evidenced in these studies, the quality and quantity of training data have escalated to pave the pathway for more sophisticated high performance pattern analysis and recognition techniques, among which *deep learning* is particularly promising and has proved to provide an effective means to learn hierarchical representations from large volumes of image data (LeCun *et al.* 2015).

Emerged from the classical machine learning domain, *deep learning* constructs learning models with multiple processing layers that have the ability of hierarchically representing features of raw data. The deep learning methods have triumphed in tackling many pattern recognition and machine learning challenges that were deemed to be difficult (LeCun *et al.* 2015). One key reason for the effectiveness of deep learning is that one complicated deep model can be properly fitted by sufficiently big data such that the diversity and variability of the training data are comprehensively characterized. One of the most popular datasets for training deep models for normal image analysis is ImageNet (Fei-Fei 2010), which consists of some fifteen million labeled images from twenty-two thousand classes. Benefited from such big data, deep learning models have shown great power in normal image analysis tasks such as detection, super-resolution, segmentation and classification.

Typical deep learning models include deep belief networks (DBNs) (Hinton *et al.* 2006) and convolutional neural networks (CNNs) (Hubel and Wiesel 1962). One DBN is a network stacked by restricted Boltzman machines that are first pre-trained in a layer-wise manner and then finely tuned through back propagation. Zou *et al.* (2015) introduced DBNs to the remote sensing community by developing a DBN based support vector machine for scene classification. Basu *et al.* (2015), from NASA Ames Research Center, proposed a statistical feature extraction method for training a DBN and compared the performance of alternative deep nets for remote sensing scene classification. On the other hand, one CNN stacks a network using interchanged convolutional filtering and pooling, which can be applied to raw images straightforwardly through extracting features hierarchically and classifying the features via the final fully-connected layer. Equipped with such advantages, CNNs have been widely used in various image analysis scenarios such as text recognition (Wang *et al.* 2012) and face identification (Sun *et al.* 2014). In order to exploit the capability of CNNs for remote sensing scene classification, Zhang *et al.* (2015) proposed a gradient boosting convolutional neural network which outperforms the scene classification schemes based on classical machine learning methodologies.

Different from existing remote sensing image classification methods which focus on improving (deep) machine learning algorithms, we investigate how to increase the diversity of training data. The volume and diversity of training data are essentially important in training a robust deep learning model. It has been observed that one deep model trained based on data with sufficient diversity tends to outperform the same model trained based on data with limited variability (Hinton *et al.* 2012). This observation reflects the necessity of data augmentation for scene classification, especially in the situation of limited available labeled remote sensing images in contrast to the increasing amount of remote sensing data. One early data augmentation method was studied in (Simard *et al.* 2003), which proposed label-preserving transformations. Recent data augmentation approaches include basic reformation of original images such as cropping andstretching (Dieleman *et al.* 2015). Krizhevsky *et al.* (2012) proposed a data augmentation method to alter intensities of the RGB channels of raw data and achieved improved performance on the ImageNet benchmark. In our work, we demonstrate how to exploit data augmentation as a preprocessing procedure for training a deep convolutional neural network and empirically evaluate the effectiveness of our data augmentation strategy for lifting the CNN representational power.

1.2. *Motivation and Contributions*

As discussed in Section 1.1, the representational power of one deep model highly relies on the diversity of training data. However, state-of-the-art deep learning strategies in remote sensing mainly focus on designing novel multi-layer representations, and are yet to investigate the impact of size and diversity of the training dataset towards their performance. The deficiency of suitable training data in remote sensing is a significant obstacle for realizing the full power of deep learning. For example, one largest labeled remote sensing dataset SAT-4 has only five hundred thousand images, and its volume sharp contrasts that of the popular normal image dataset ImageNet which contains fifteen million labeled images from twenty-two thousand classes.

This paper aims to address these fundamental data limitations that hinder the maximization of deep learning's power in remote sensing image classification. We introduce a methodology to enhance the volume and diversity of remote sensing datasets, and exploit the enhanced datasets to train a deep convolutional neural network. In order to maintain the key original feature representations and avoid distortions, we carefully select three basic data augmentation operations including flip, translation and rotation, which not only significantly enhance the size and completeness of the dataset, but also preserve the scene topologies. We further investigate the application of the enhanced dataset in training one deep convolutional neural network and the impact of its performance in remote sensing scene classification. In our implementation, the augmentation approach adopts the set of simple operations with low computational complexity, and is performed on the central processing unit (CPU), while the training is conducted on Graphics Processing Units (GPUs). Our contributions to the remote sensing literature are two-fold. Methodologically, we introduce a data augmentation strategy to diversify the training dataset, lifting the representation power of normal convolutional neural networks for scene classification. Empirically, experimental results achieve state-of-the-art performance on benchmark remote sensing imagery datasets and outperform existing scene classification deep models.

The rest of this paper is organized as follows. Section 2 introduces the data augmentation enhanced deep learning methodology. Section 3 validates the proposed methodology in scene classification using three practical benchmark remote sensing datasets. Section 4 concludes this paper.

2. The Data Augmentation Enhanced Deep Learning Framework

In this section, we introduce the data augmentation enhanced deep learning methodology for remote sensing scene classification. We first describe the basic data augmentation operations for enhancing the volume and diversity of a dataset, then present a deep convolutional neural network exploiting the enhanced dataset, and finally explain how to train the network using the augmented dataset.

2.1. *Data Augmentation*

The representational power of machine learning models (especially deep learning models) highly relies on the training procedures by using plenty of diverse training data. However, though the amount of remote sensing data keeps increasing every year, the properly labeled remote sensing images available for training a deep machine learning model are still limited. We describe how to enhance existing remote sensing datasets via data augmentation and produce augmented datasets to train a more robust deep convolutional neural network.

Data augmentation aims at generating additional and more diversified data samples through certain transformations conducted upon original data. In our work, we use publicly available remote sensing scene image sets as the original datasets. For a given remote sensing scene image set $\mathcal{D}_o = \{I_1, \cdots, I_K\}$, where I_k indicates the kth image sample in the dataset. Suppose that I_k has totally N pixels. The pixel homogeneous coordinate matrix $\mathcal{P}_k$ for I_k is

$$\mathcal{P}_k = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}, \tag{1}$$

where each row represents the homogeneous coordinate for one pixel.

The data augmentation operation on one image I_k is to apply an affine transformation matrix $\mathcal{M}$ to its homogeneous coordinate matrix $\mathcal{P}_k$ and obtain a transformed homogeneous coordinate matrix $\mathcal{P}_k^t$ for the image. The operation is presented as follows:

$$\mathcal{P}_k^t = \mathcal{P}_k \mathcal{M}. \tag{2}$$

Here each row of $\mathcal{P}_k^t$ is the transformed homogeneous coordinate for one pixel.

There are various ways to determine the affine transformation matrix $\mathcal{M}$. In our work, we use three types of random perturbations for generating new augmented data. The three types of transformations are described as follows:

- **Flip** (denoted as $\mathcal{T}_1$): The image is flipped along the horizontal dimension. The corresponding affine transformation matrix $\mathcal{M}$ is shown in the 'Flip' column of Table 1.
- **Translation** (denoted as $\mathcal{T}_2$): The image is shifted in both the x and y directions of the image. The corresponding affine transformation matrix $\mathcal{M}$ is shown in the 'Translation' column of Table 1. T_x and T_y are the offsets on the coordinate axis.

- **Rotation** (denoted as $\mathcal{T}_3$): The image is rotated with an angle sampled from 0° to 180°. The corresponding affine transformation matrix $\mathcal{M}$ is shown in the 'Rotation' column of Table 1, where β is the rotation angle.

Table 1.: Transformation Matrices for augmentation operations.

Operations	Flip	Translation	Rotation
Transform Matrix	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$	$\begin{bmatrix} cos\beta & -sin\beta & 0 \\ sin\beta & cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Given one image I_k, the augmented data is denoted as $O_k = \{\mathcal{T}_1(I_k), \mathcal{T}_2(I_k), \mathcal{T}_3(I_k)\}$. The augmented dataset for the original dataset $\mathcal{D}_o$ is denoted as $\mathcal{D}_a = \{O_1, \cdots, O_K\}$. The augmentation process of the dataset $\mathcal{D}_o$ is thus formulated as follows:

$$\mathcal{D}_a = \{O_1, \cdots, O_K\} = \bigcup_{k=1}^{K} \bigcup_{i=1}^{3} \mathcal{T}_i(I_k). \tag{3}$$

The augmented dataset $\mathcal{D}_a$, along with the corresponding class labels, is then used to train a convolutional neural network for the purpose of remote sensing image classification.

It should be noted that we exploit flips, translations and rotations as the data augmentation operations because they do not change the scene topologies in remote sensing imageries, which are essentially important for consistent scene classification. These operations do not increase the spectral or topological information for the data. However, these preprocessing operations help the deep model capture the data intrinsics more comprehensively. For example, a common machine learning model cannot necessarily categorize the scene image, which is just an upside-down instance of an image from the training dataset, into its correct class. One reason for this deficiency is that the upside-down scene image, though with no more information over its original image, is not involved in the training procedure such that the trained machine tended not to 'know' its class label. This example reflects the importance of data diversity for training a robust classification machine. One advantage of deep convolutional neural networks over common machine learning methods is their greater capability of characterizing the immense diversity of big data. The augmentation operations such as flips, translations and rotations increase the diversity of training data. The representational power of a deep convolutional neural network is thus greatly improved through being trained based on augmented data than that trained without data augmentation.

The same data augmentation operations do not necessarily improve the classification accuracy for common machine learning methods, which usually are 'shallowly' structured and tend to have weaker capability than deep models in terms of characterizing data diversity.

2.2. *Convolutional Neural Network*

Convolutional neural networks (CNNs) are a type of feed-forward artificial neural networks (ANNs), with the multilayer structure consisting of convolutional layers, pooling

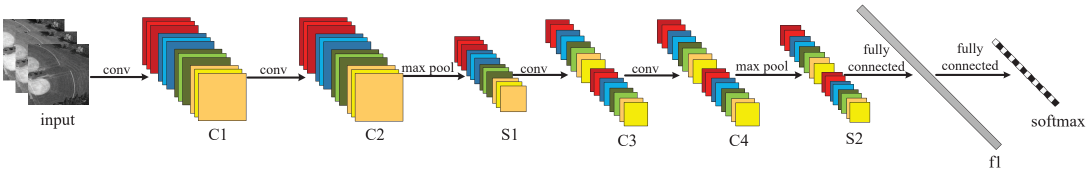


Figure 1.: The CNN architecture for remote sensing scene classification.

layers and fully-connected layers. We use the CNN structure illustrated in Figure 1 for remote sensing scene classification. Here a remote sensing image is one input of the convolutional neural network. C1, C2, C3 and C4 are four convolutional layers. Each convolutional layer consists of feature maps generated by applying convolutional kernels to convolving the previous layers. S1 and S2 are two max-pooling layers, respectively. The max-pool operation maps the maximum within one local region to one number. One max-pooling layer consists of feature maps obtained from applying max-pool operations to previous layer feature maps. Different colors in Figure 1 indicate that feature maps generated with different convolutional kernels. The CNN is finalized by a fully-connected layer and a softmax layer which outputs the classification result. We refer the interested readers to the landmark work (Simonyan and Zisserman 2014) for CNN construction details.

2.3. *Training A Convolutional Neural Network*

Training a convolutional neural network is to compute the optimal parameter values for the network based on a training set, i.e., labeled remote sensing images as inputs and their corresponding labels as target outputs. In our work, we use the augmented dataset $\mathcal{D}_a$ described in Section 2.1 as the training inputs. One input image is processed through sequential interchanged convolution and max-pooling layers. Each convolution layer generates feature maps and each max-pooling layer downsizes the feature maps in terms of the neighboring maximization pooling strategy. A convolution and max-pooling couple is illustrated in Figure 2:

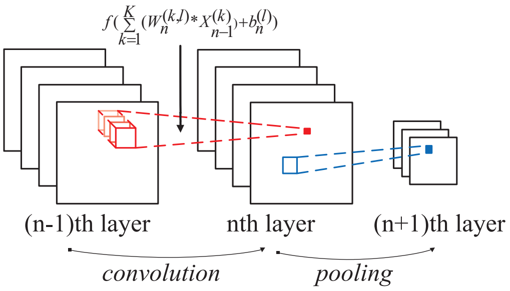

$f(\sum_{k=1}^{K}(W_n^{(k,l)} * X_{n-1}^{(k)}) + b_n^{(l)})$
(n-1)th layer
nth layer
(n+1)th layer
convolution
pooling

Figure 2.: Convolution and pooling layers.

The layers C1, C2 and S1 (or similarly C3, C4 and S2) in Figure 1 can be thought of examples of the $(n-1)$th, nth and $(n+1)$th layers in a concrete CNN, respectively.

Specifically, the feature maps for the nth layer, which is a convolution layer, are obtained by convolving the $(n-1)$th layer with trainable parameters (i.e., the weight W_n and bias b_n) and then processed by an activation function $f(\cdot)$. The trainable parameters are initialized randomly subject to a uniform distribution. The convolution and activation operations (marked in red in Figure 2) result in the feature maps for the nth layer

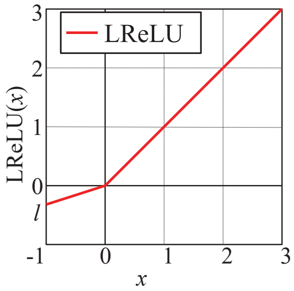


Figure 3.: LReLU.

as follows:

$$X_n^{(l)} = f(\sum_{k=1}^{K}(W_n^{(k,l)} * X_{n-1}^{(k)}) + b_n^{(l)}). \tag{4}$$

Here the activation function $f(\cdot)$ is a leaky rectified linear unit (LReLU) (Maas *et al.* 2013) as illustrated in Figure 3, and $*$ indicates the convolution operation.

The K feature maps for the $(n-1)$th layer is represented as a set of matrices $X_{n-1}^{(k)}$ $(k = 1, 2, \cdots, K)$, and the lth feature map for the nth layer is represented as $X_n^{(l)}$. A max-pooling layer (e.g. the $(n+1)$th layer in Figure 2) maps maxima within local regions of the previous layer to individual numbers (marked in blue in Figure 2). The feature maps are finally convolved with fully connected layers and generate a target output (i.e. a predicted class label). We measure the error between the target output and the true label of the input remote sensing image. The trainable parameters for each layer are optimized subject to the error minimization. The optimization of trainable parameters is achieved by an effective back propagation method referred to as the mini-batch stochastic gradient descent (SGD) (Ngiam *et al.* 2011). We repeatedly optimize the trainable parameter throughout all augmented training data and obtain the trained network. Specifically, the network utilized in our work is one simplified version of VGGNet (Simonyan and Zisserman 2014) with a structure of eight layers (as illustrated in Figure 1). To prevent overfitting in training the CNN, a dropout operation (Srivastava *et al.* 2014) is applied after every max-pooling layer. Specifically, the implementation of dropout involves randomly choosing a certain number of neurons (i.e. activation function units) during each training step and performing back propagation only through them. Dropout is regularization technique for training networks, which prevents convolutional neural networks from overfitting (Srivastava *et al.* 2014).

The trained network is then used for classifying unlabeled remote sensing images by predicting class labels for them. For an unlabeled remote sensing image, we predict its class label by using the trained convolution neural network with the optimal parameters. In testing, the convolution, activation and pooling are performed in similar ways as those in training. The only difference is that the parameters such as W_n and b_n in (4) are adjustable in training. However, they are fixed optimal values in testing. It is in such a way that the label of an unknown remote sensing image is predicted.

3. Experimental Evaluations

In this section, we empirically evaluate our strategy for training a deep convolutional neural network for scene classification based on data augmentation. We first introduce the benchmark remote sensing image datasets, then describe the experiment settings, and finally present the experimental results of alternative methods on the benchmark datasets.

3.1. *Datasets*

Three benchmark remote sensing datasets are used for experimental evaluations. The first dataset is SAT which contains two subdatasets, i.e. SAT-4 and SAT-6 datasets[1]. The second dataset is RSSCN7[2]. The third dataset is UC Merced Land Use[3].

Both SAT-4 and SAT-6 were extracted from the NASA National Agriculture Imagery Program (NAIP) dataset. Specifically, SAT-4 consists of 500,000 images which cover four scene classes, i.e. barren lands, trees, grasslands and a class involving various scenes other than the above three. SAT-6 consists of 405,000 images which cover six scene classes, i.e. barren lands, trees, grasslands, roads, buildings and water bodies. The resolution for individual images in SAT-4 and SAT-6 is 28×28. The RSSCN7 dataset contains 2,800 remote sensing images which are from seven typical scene categories, i.e. grass lands, forests, farm lands, parking lots, residential regions, industrial regions and water bodies. For each category, there are 400 images sampled on four different scales with 100 images per scale. The resolution of individual images is 400×400. The RSCNN7 dataset is rather challenging due to the wide differences of the scene images which were captured under changing seasons and varying weathers, and sampled with different scales. The UC Merced Land Use dataset (abbreviated as UCMerced) is a popular dataset for remote sensing scene classification and contains 2,100 images which are from 21 scene categories. For each category, there are 100 images. The resolution for individual images is 256×256. Images in UCMerced were manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around the country.

3.2. *Experiment Settings*

To validate the effectiveness of the data augmentation, we design two sets of experiments for training one common convolutional neural network. The first set of experiments use the original datasets to train the CNN. The second set of experiments first perform the data augmentation to the original datasets and then use the augmented datasets to train the same CNN. We refer to the first and second set of experiments as no-aug-experiments and aug-experiments, respectively. For aug-experiments, we configure the augmentation parameters and the CNN training hyper-parameters by searching optimal parametric values from a grid set of candidate parametric values, which includes three rotation degree candidates, three width shift ratio candidates, three height shift ratio candidates, and two learning rate candidates. The candidate parametric values for grid search with respect each dataset are described in Table 2. The mathematical relations between the augmentation parameters and the parameters in the transformation matrices in Table 1

[1] http://csc.lsu.edu/˜saikat/deepsat/
[2] https://sites.google.com/site/qinzoucn/documents
[3] http://vision.ucmerced.edu/datasets/landuse.html

are presented as follows:

$$\begin{cases} \beta = \pi * \mathcal{U}(-\text{Rotation}, \text{Rotation})/180 \\ T_y = \mathcal{U}(-\text{Width shift}, \text{Width shift}) * W \\ T_x = \mathcal{U}(-\text{Height shift}, \text{Height shift}) * H \end{cases} \tag{5}$$

where $\mathcal{U}(a, b)$ denotes a random value sampled subject to the uniform distribution in the interval $[a, b]$, and W and H denote the width and height of one image, respectively.

Table 2.: Candidate parametric values for grid search.

Datasets	Augmentation parameters			CNN hyper-parameters
	Rotation (Degree)	Width shift (Ratio)	Height shift (Ratio)	Learning Rate (Value)
SAT	10, 20, 30	0.1, 0.2, 0.3	0.1, 0.2, 0.3	0.005, 0.010
RSSCN7	10, 20, 30	0.1, 0.2, 0.3	0.1, 0.2, 0.3	0.001, 0.005
UCMerced	10, 20, 30	0.1, 0.2, 0.3	0.1, 0.2, 0.3	0.001, 0.005

Table 3.: Dataset organization.

Datasets	Subsets (Number of images)			Image resolution	
	Training	Validation	Test	Height	Width
SAT-4	2800	1199	1000	28	28
SAT-6	2268	971	810	28	28
RSSCN7	1400	840	560	64	64
UCMerced	1050	630	420	64	64

It is clear that we totally have 54 sets of parametric configurations for training a CNN based on an augmented dataset. In order to observe the effects of data augmentation for training a CNN model, we split subsets of SAT-4 and SAT-6 and the whole datasets RSSCN7 and UCMerced into training, validation and test subsets separately, as described in Table 3. For each dataset, we conduct 54 individual aug-experiments and one no-aug-experiment. Especially, each of the 54 individual aug-experiments is set subject to a combination of parametric candidates (i.e. three rotation candidates, three width shift candidates, three height shift candidates and two learning rate candidates). The experimental results are compared in terms of Kappa Index (i.e. Cohen's kappa) (Hrechak and Mchugh 1990, Cohen 1960).

The data augmentation operations are performed on CPU, and the CNN training procedures are conducted on an NVIDIA GeForce GTX TITAN X 12 GB GPU. For all aug-experiments, we trained the deep models with the same architecture as described in Sections 2.2 and 2.3 based on the augmented datasets, and for all no-aug-experiments, we train the same deep models based on the original datasets.

3.3. *Experimental Results*

We train a convolutional neural network based on each of the four datasets, i.e. SAT-4, SAT-6, RSSCN7 and UCMerced, separately. For each dataset, both aug-experiments and no-aug-experiments are conducted. The parametric values for the aug-experiments are set according to Table 2. We thus have 54 different sets of parametric values for training a CNN using one augmented dataset. The testing results in terms of Kappa index are shown in Table 4. The 'No-aug' column gives the Kappa indices for testing the CNN trained by using original datasets. On the other hand, each entry of the 'Aug' column gives a Kappa index range for testing the CNN trained by using the augmented dataset subject to 54 different sets of parametric configurations. It is clear that for each dataset, even the smallest Kappa index for aug-experiments is greater than that of no-aug-experiments. We perform grid search to obtain optimal parametric configurations, which are shown in Table 5. These experiments validate that one CNN trained by using augmented remote sensing dataset outperforms that trained by using the original remote sensing dataset.

Table 4.: All Results of No-aug-experiment and aug-experiments

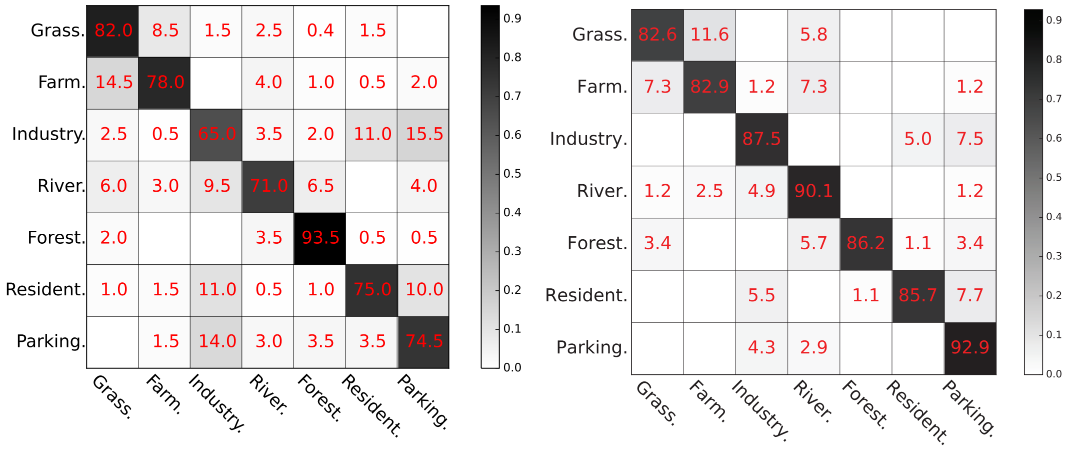
Datasets	Kappa Index	
	No-aug	Aug
SAT-4	0.83	0.87 - 0.96
SAT-6	0.94	0.94 - 0.97
RSSCN7	0.61	0.71 - 0.85
UCMerced	0.48	0.71 - 0.87

Table 5.: Optimal parametric settings for aug-experiments

Datasets	Aug Params			CNN Hyper-params	kappa Index
	Rotation	Width Shift	Height Shift	Learning Rate	
SAT-4	10	0.3	0.1	0.005	0.96
SAT-6	10	0.3	0.3	0.005	0.97
RSSCN7	30	0.3	0.2	0.005	0.85
UCMerced	10	0.2	0.3	0.005	0.87

To make the empirical evaluation one step further, we experimentally compared our framework with two state-of-the-art deep learning based scene classification approaches.

We first test our model with the augmented data and compare it with the state-of-the-art deep feature selection model (Zou *et al.* 2015) using the RSSCN7 dataset. We use 50% of the data for training our CNN and use the 'Test' subset for testing the classification performance of our trained model. Our method achieves an kappa index of 0.85, which is better than the 0.73 obtained by the deep feature selection model (Zou *et al.* 2015). Figure 4 shows the comparison of the confusion matrices of the classification results achieved by the deep feature selection model and our method. It is clear that the classification accuracy of our model outperforms the state-of-the-art deep feature selection model on all categories except *forest*. Here we observe that the classification accuracy of the deep



(a) (b)

Figure 4.: Confusion matrix: (a) Zou *et al.* (2015) and (b) Ours.

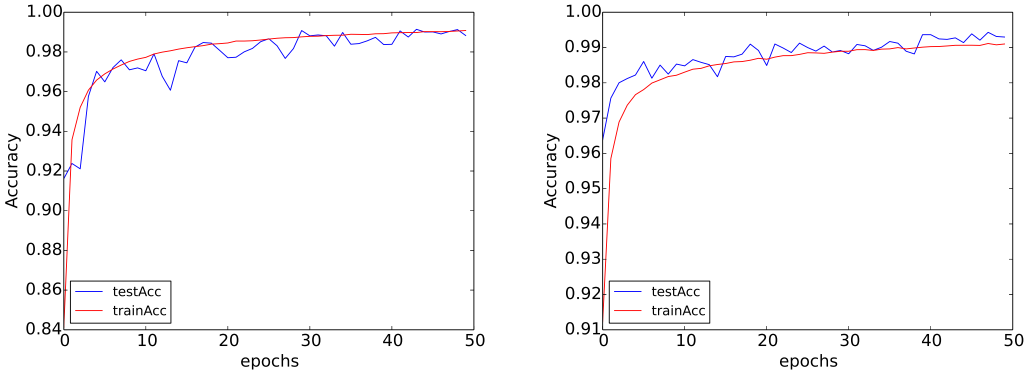
Table 6.: Classification Accuracy(%).

Method	SAT-4	SAT-6
DeepSat (Basu *et al.* 2015)	97.946	93.916
Our Method	99.127	99.297

feature selection model has large variation over different categories. On the other hand, the classification accuracy of our model in each category is not substantially different from the rest. This result reveals that our method not only achieves better overall classification performance but also exhibits greater robustness than the deep feature selection model.

Then, we experimentally test our model based on the datasets SAT-4 and SAT-6. In order to evaluate the fitness of our scene classification model, we draw learning curves during the whole training and testing procedures with 50 epochs. Figure 5 illustrates the training accuracy and testing accuracy of the CNN on the SAT-4 and SAT-6 datasets. We observe that the training curve and test curve fit well with each other. This implies that the data augmentation operations enable us to train a deep model with a reasonable balance between the variance in training and the bias in testing.

In addition, we compare our model with a state-of-the-art deep learning framework DeepSat (Basu *et al.* 2015), which utilizes 22 features selected from 150 extracted features using feature-ranking to train a DBN classifier. To make a fair comparison, we follow the same experimental setting with DeepSat for our method and do not use our own experimental setting described above. The results of DeepSat and our method are shown in Table 6. The classification accuracy of our framework on SAT-4 and SAT-6 datasets reaches 99.127% and 99.297%, respectively, both of which are better than the results obtained by the DeepSat framework.



(a)

(b)

Figure 5.: Training and testing accuracy curves: (a) SAT-4 and (b) SAT-6.

4. Conclusions

This paper introduces basic data augmentation operations to address the fundamental data limitation problem in applying deep learning for remote sensing image processing. We describe how to use data augmentation to improve the remote sensing scene classification performance of convolutional neural networks. We show that the diversity and completeness of data can be greatly enhanced by data augmentation, and when applied to training deep learning models, the experimental results with augmentation operations outperform those from the same deep model architecture training on the original dataset. The effectiveness and robustness of our proposed methodology are confirmed by experiments using practical remote sensing datasets. The proposed methodology advances the state-of-the-art and can significantly contribute to the new horizon of deep learning in remote sensing.

Though the data augmentation strategy enhances the diversity of the dataset to a certain extent, it just increases the visual variability of each training remote sensing image subject to its intrinsic spectral and topological constraints and does not generate new information for the remote sensing image. Future research will focus on exploiting state-of-the-art generative adversarial nets to generate new remote sensing image instances based on a trained augmentation strategy beyond the basic flip, translation and rotation operations.

4.1. *Acknowledgements*

This work was supported in part by National Natural Science Foundation of China (No. 61671481), Qingdao Applied Fundamental Research (No. 16-5-1-11-jch) and the Fundamental Research Funds for Central Universities.

References

Basu, S., *et al.*, 2015. DeepSat-A learning framework for satellite imagery. *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 37:1–37:10.

Cohen, J., 1960. A coefficient of agreement for nominal scales. *Educational & Psychological Measurement*, 20 (1), 37–46.

Cui, S., Schwarz, G., and Datcu, M., 2015. Remote sensing image classification: No features, no clustering. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8 (11), 5158–5170.

Dieleman, S., Willett, K.W., and Dambre, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450 (2), 1441–1459.

Fei-Fei, L., 2010. ImageNet: crowdsourcing, benchmarking & other cool things. *CMU VASC Seminar.*

Hinton, G., *et al.*, 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29 (6), 82–97.

Hinton, G., Osindero, S., and Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18 (7), 1527–1554.

Hrechak, A.K. and Mchugh, J.A., 1990. Automated fingerprint recognition using structural matching. *Pattern Recognition*, 23 (8), 893–904.

Hubel, D.H. and Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160 (1), 106–110.

Krizhevsky, A., Sutskever, I., and Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25 (2).

LeCun, Y., Bengio, Y., and Hinton, G., 2015. Deep learning. *Nature*, 521 (7553), 436–444.

Maas, A.L., Hannun, A.Y., and Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. *International Conference on Machine Learning.*

Ngiam, J., *et al.*, 2011. On optimization methods for deep learning. *International Conference on Machine Learning*, 67–105.

Simard, P.Y., Steinkraus, D., and Platt, J.C., 2003. Best practices for convolutional neural networks applied to visual document analysis. *International Conference on Document Analysis and Recognition*, 2, 958–962.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *Computer Science.*

Srivastava, N., *et al.*, 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15 (1), 1929–1958.

Sun, Y., Wang, X., and Tang, X., 2014. Deep learning face representation from predicting 10,000 classes. *IEEE Conference on Computer Vision and Pattern Recognition*, 1891–1898.

Tang, Y. and Pannell, C.W., 2009. A hybrid approach for land use/land cover classification. *GIScience & Remote Sensing*, 46 (4), 365–387.

Tomas, L., *et al.*, 2016. Urban population estimation based on residential buildings volume using IKONOS-2 images and lidar data. *International Journal of Remote Sensing*, 37 (sup1), 1–28.

Wang, T., *et al.*, 2012. End-to-end text recognition with convolutional neural networks. *International Conference on Pattern Recognition*, 3304–3308.

Zhang, F., Du, B., and Zhang, L., 2015. Scene classification via a gradient boosting random convolutional network framework. *IEEE Transactions on Geoscience and Remote Sensing*, 30 (99), 1–10.

Zhou, H. and Gao, H., 2014. Fusion method for remote sensing image based on fuzzy integral. *Journal of Electrical and Computer Engineering*, 2014, 26–34.

Zou, Q., *et al.*, 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12 (11), 2321–2325.