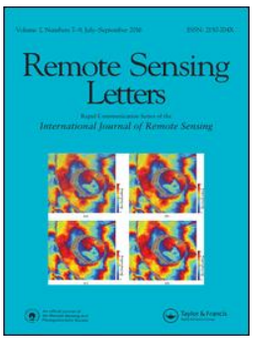


Taylor & Francis
Taylor & Francis Group

Remote Sensing Letters

ISSN: 2150-704X (Print) 2150-7058 (Online) Journal homepage: http://www.tandfonline.com/loi/trsl20

SatCNN: satellite image dataset classification using agile convolutional neural networks

Yanfei Zhong, Feng Fei, Yanfei Liu, Bei Zhao, Hongzan Jiao & Liangpei Zhang

To cite this article: Yanfei Zhong, Feng Fei, Yanfei Liu, Bei Zhao, Hongzan Jiao & Liangpei Zhang (2017) SatCNN: satellite image dataset classification using agile convolutional neural networks, Remote Sensing Letters, 8:2, 136-145, DOI: 10.1080/2150704X.2016.1235299

To link to this article: http://dx.doi.org/10.1080/2150704X.2016.1235299

REMOTE SENSING LETTERS, 2017
VOL. 8, NO. 2, 136–145
http://dx.doi.org/10.1080/2150704X.2016.1235299

Taylor & Francis
Taylor & Francis Group

SatCNN: satellite image dataset classification using agile convolutional neural networks

Yanfei Zhong[a], Feng Fei[a], Yanfei Liu[a], Bei Zhao[b], Hongzan Jiao[c] and Liangpei Zhang[a]

[a]State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, China; [b]Department of Geography and Resource Management, The Chinese University of Hong Kong, Hong Kong, China; [c]School of Urban Design, Wuhan University, Wuhan, China

ABSTRACT

With the launch of various remote-sensing satellites, more and more high-spatial resolution remote-sensing (HSR-RS) images are becoming available. Scene classification of such a huge volume of HSR-RS images is a big challenge for the efficiency of the feature learning and model training. The deep convolutional neural network (CNN), a typical deep learning model, is an efficient end-to-end deep hierarchical feature learning model that can capture the intrinsic features of input HSR-RS images. However, most published CNN architectures are borrowed from natural scene classification with thousands of training samples, and they are not designed for HSR-RS images. In this paper, we propose an agile CNN architecture, named as SatCNN, for HSR-RS image scene classification. Based on recent improvements to modern CNN architectures, we use more efficient convolutional layers with smaller kernels to build an effective CNN architecture. Experiments on SAT data sets confirmed that SatCNN can quickly and effectively learn robust features to handle the intra-class diversity even with small convolutional kernels, and the deeper convolutional layers allow spontaneous modelling of the relative spatial relationships. With the help of fast graphics processing unit acceleration, SatCNN can be trained within about 40 min, achieving overall accuracies of 99.65% and 99.54%, which is the state-of-the-art for SAT data sets.

ARTICLE HISTORY
Received 11 February 2016
Accepted 6 September 2016

1. Introduction

Satellite image data sets have the characteristics of a big data volume and complex image classification. With the increase in the spatial resolution of on-board sensors, the diversity of intra-class objects is increasing, as is the similarity of inter-class objects (Zhao et al. 2016b). High-spatial resolution remote-sensing (HSR-RS) images allow more pixels to describe objects and their surroundings, which automatically build up higher level information. In the HSR-RS image, the patches with comparatively clear semantics are commonly referred to as 'semantic' scenes (Hare et al. 2006). Scene classification is aimed at building a reasonable and robust mapping from raw image scenes to their correct high-level semantics. However, by merely depending on empirical or simple low-

CONTACT Hongzan Jiao jiaohongzan@whu.edu.cn School of Urban Design, Wuhan University, Wuhan, China

level features, such as simple statistics of raw pixel values or simple image descriptors, the traditional classification algorithms have a limited ability in interpreting meanings of scenes.

Recently, mid-level features have been proved to be an effective bridge between raw image pixels and image meanings in high-level human recognition. The mid-level features denote those intermediate properties extracted from low-level local features, which selectively represent more information that is helpful in scene classification. Currently, there are two main ways to build up mid-level features to describe the semantic of an image (Bosch, Muñoz, and Mart 2007): (1) feature engineering, for instance, the bag-of-visual-word model (Lazebnik, Schmid, and Ponce 2006) or topic models such as probabilistic latent semantic analysis (Zhou, Zhou, and Hu 2013) and latent Dirichlet allocation (Zhao, Zhong, and Zhang 2013; Zhong, Zhu, and Zhang 2015; Zhao et al. 2016a); and (2) feature learning in which features are automatically learned from an optimization problem solved by machine learning algorithms, for example, sparse coding (Cheriyadat 2012), unsupervised feature learning (Zhang, Du, and Zhang 2015) and deep learning (DL). DL is the recent breaking through of machine learning, concentrated on discovering intrinsic features with deeper hierarchical neural network models (Zhou et al. 2015; Zhao et al. 2015). Convolutional neural networks (CNNs) are one of the most popular DL models, which are partly based on a model of simple and complex cells in the primary visual cortex and consist of convolutional layers with optional pooling layers. CNNs are usually supervised models, which can yield outstanding performances in natural image classification, object detection, speech recognition and face recognition (LeCun, Bengio, and Hinton 2015).

As for HSR-RS scene patches, CNN is a scene classification approach based on deep learning, which focuses on bridging the low-level features to high-level semantics of the image scene and can automatically extract intrinsic features from HSR-RS imagery. In HSR-RS scene classification, Castelluccio et al. (2015) discussed transferring CNN to an HSR-RS data set; Hu et al. (2015) reported improved performance of transferred deep CNN features. These studies were all based on small HSR-RS data sets such as the UC Merced land-use data set. Basu et al. (2015) reported on the performance of simple DL models (including CNNs) on HSR-RS scene data sets, called SAT data sets, that have more than 300,000 image samples for training. However, except for the transfer learning of CNN models, previous CNN architectures have either been largely borrowed from natural image scene classification or direct copies of the original CNN architectures. In these architectures, the fully connected network (FCN) layers occupy approximately 80% of the total parameters of CNN, which result in a great danger of overfitting without enough training samples. Although transfer learning has achieved surprising performances, the exploration of reasonable and efficient CNN architecture directly trained from larger HSR-RS scene databases has not yet been reported.

This paper introduces a deep CNN architecture for HSR-RS scene data sets. The major contributions of this paper are as follows:

(1) Based on published deep CNN applications in natural scene classification, an agile CNN architecture for HSR-RS scene classification, named SatCNN, is proposed by

balancing the size between HSR-RS scene input and receptive fields of feature vectors in classifier layers.

(2) Given different setting of hyper-parameters in the proposed SatCNN scene classification framework, the scene classification accuracy and efficiency are comprehensively analysed. In addition, with the help of graphics processing unit (GPU) acceleration, the whole framework can achieve good performances within about 40 min.

Experiments on SAT data sets (Basu et al. 2015) confirmed that the SatCNN models can effectively learn reliable and robust features for large-volume satellite image scene classification.

2. SatCNN scene classification framework

The general flowchart of the SatCNN framework is shown in Figure 1. The HSR-RS scene image **X** is first normalized before learning hierarchical features using convolution, activation and pooling. After feature learning, the output feature vector is obtained and classified by Softmax. The parameters of CNN model can be updated using the back-propagation and the loss function $J(\cdot)$. During the CNN model training and classifying, the convolutional layers need a great amount of calculation, which can be greatly accelerated by GPU computation. The details of the procedure are described as the following parts.

2.1. *Agile CNN architecture*

The designs of the published CNN models are highly correlated to their input data source. In order to be an agile model for the HSR-RS scenes, the architectures of the deep CNN models need to be improved via balancing the depth of the CNN architectures and the number of model parameters according to the characteristics of the HSR-RS images.

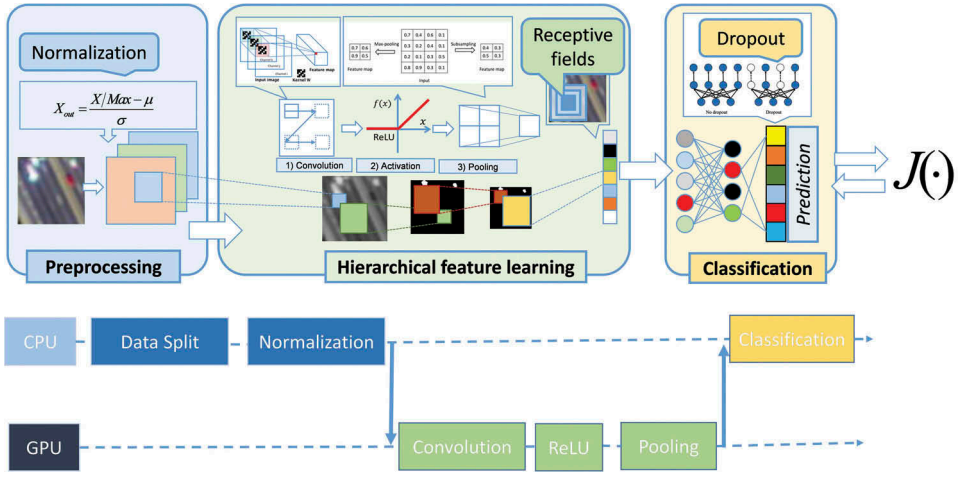


Figure 1. General flowchart of the SatCNN scene classification framework.

Table 1. Receptive fields of output feature representation vectors in CNN models with different depths.

Model	Depth	Receptive fields
5 × 5c-2p	2	10 × 10
3 × 3c-2p-3 × 3c-2p	4	36 × 36
5 × 5c-2p-5 × 5c-2p	4	100 × 100
3 × 3c-2p-3 × 3c-2p-3 × 3c-2p	6	216 × 216
5 × 5c-2p-5 × 5c-2p-5 × 5c-2p	6	1000 × 1000

Instead of global connection in multilayer perceptron (MLP), deep CNNs restrict the connection to be local by the use of spatial convolution kernels, the size of which is commonly set as 3 × 3 pixels or 5 × 5 pixels (Simonyan and Zisserman 2014). Table 1 reports the receptive fields of feature vectors with different depths, where the value before c is the size of convolution kernel, the value before p is the stride in pooling layer. According to Table 1, the receptive fields of the feature vectors become larger as the layers becoming deeper. If the depth of a CNN model is not sufficient, the output receptive fields of the feature representation cannot capture the global features.

2.2. *Modern CNN improvements used in SatCNN*

2.2.1. *Image and feature normalization*

Generally, the digital values in real HSR-RS images are integers. However, neural network parameters and activation functions are randomly initialized between $[0, 1]$, which results in that normalization is essential to avoid abnormal gradients. The *z*-score can adjust the distribution of pixel values in input images to an approximately normal distribution, which facilitates the activation and gradient descent progress (LeCun et al. 2012). The *z*-score normalization can be written as the following equation.

$$\mathbf{X}_{\text{out}} = \frac{(\mathbf{X}/\text{Max}) - \mu}{\sigma} \tag{1}$$

where Max is the maximum pixel value in the image, μ and σ are the mean and standard deviation of $\mathbf{X}$/Max, respectively, and $\mathbf{X}_{\text{out}}$ is normalized data.

2.2.2. *The rectified linear unit and dropout*

The rectified linear unit (ReLU) is a simple function not only in calculating its output, but also in calculating its first derivative (Krizhevsky, Sutskever, and Hinton 2012). As for HSR-RS images, the ReLU selects the 'foreground' of the convolutional feature map and discards values smaller than 0 as 'background'. Furthermore, in order to avoid over-fitting, the SatCNN model uses a dropout technique to perform regularization, which randomly drops out units (along with their connections) from neural network during training (Srivastava et al. 2014). The combination of ReLU and dropout forces the convolution kernels to capture more robust features for the scene representation.

2.2.3. *Cross-entropy loss function*

We let $\mathbf{c}$ stand for the feature vector extracted from CNN, that is, $c = f(\mathbf{X}_{\text{out}}, \mathbf{W}, b)$ where f is the feature extraction function, $\mathbf{W}$ and b are parameters of CNN. In CNN, the

prediction vector $\mathbf{y}^i = (y_1^i, y_2^i, \ldots, y_k^i)$ is got by Softmax classifier, where y_t^i represents the possibility of the ith sample's label being t and is computed by Equation (2), where θ represents the parameters in Softmax layer, k is the number of labels.

$$y_t^i = \frac{\exp(\theta_t^T \mathbf{c})}{\sum_{j=1}^{k} \exp\left(\theta_j^T \mathbf{c}\right)} \tag{2}$$

The loss function of CNN can be written as Equation (3), where N is the number of samples, and $\mathbf{1}\{\cdot\}$ is the indicator function.

$$J(\mathbf{X}_{\text{out}}, \mathbf{W}, b, \theta) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=1}^{k} \mathbf{1}\{y^i = t\} \cdot y_t^i\right] \tag{3}$$

2.2.4. *Stochastic gradient descent*

Stochastic gradient descent (SGD) uses the mini-batch's back propagation error to approximate the error of all the training samples, which accelerates the cycle of the weight update with smaller back propagation error to speed up the convergence of the whole model. During back propagation, the Equations (4), and (5) are adopted to update $\mathbf{W}$ and b in every layer, where λ is the momentum which help accelerate SGD by adding a fraction of the update value of the past time step to the current update value, α is the learning rate, $\nabla\mathbf{W}$ and ∇b are the gradients of $J(\cdot)$ w.r.t. $\mathbf{W}$ and b, respectively, t just stands for the number of epoch during SGD:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \mathbf{V}_{t+1}, \quad \mathbf{V}_{t+1} = \lambda\mathbf{V}_t + \alpha\nabla\mathbf{W} \tag{4}$$

$$b_{t+1} = b_t - U_{t+1}, \quad U_{t+1} = \lambda U_t + \alpha\nabla b \tag{5}$$

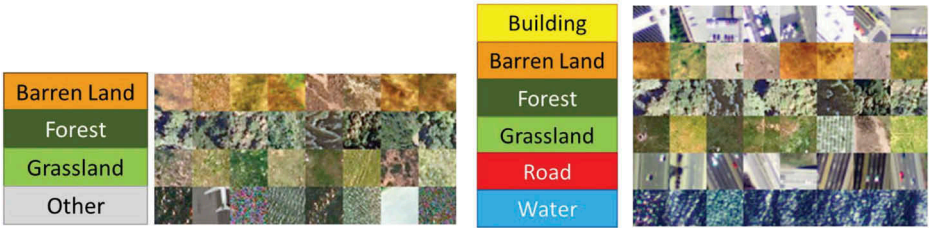
2.3. *Fast GPU acceleration*

Spatial convolution is the major computation operation of the SatCNN model, which needs the solution to be accelerated. The NVIDIA provides deep neural network library (CuDNN) of compute unified device architecture (CUDA) to support for deep architecture optimized training by integrating hardware architecture and software acceleration. Therefore, in the proposed approach, a hybrid computation style based on CuDNN is used as shown in Figure 1: most of the ancillary operations, such as sample shuffling, image normalization and data transfer between device and host, are performed in the CPU, while the GPU focuses on CNN model training.

3. Experiments

3.1. *Experiment setup*

The SAT-4 and SAT-6 data sets are used to evaluate SatCNN. The SAT-4 data set contains 500,000 image patches, including four classes: barren land, trees, grassland and a class



(*a*) SAT-4 land-use dataset (*b*) SAT-6 land-use dataset

Figure 2. Image samples from two SAT data sets. (*a*) SAT-4 data set, (*b*) SAT-6 data set.

that consists of all other land cover classes (Figure 2(*a*)). SAT-6 is a data set of 405,000 image patches covering 6 land-cover types, including barren land, trees, grassland, roads, buildings and water bodies (Figure 2(*b*)). The image patches in both data sets are with the size of 28 × 28 pixels and 1-m spatial resolution.

The SatCNN architecture was built up with three convolutional layers followed by a two layer FCN classifier and this architecture yielded 64 feature maps in the first convolutional layer. The convolutional kernels were kept as 3 × 3 and the feature pooling window was dynamically adjusted to ensure that all the pixels in the feature maps could contribute to the next layer. Dropout was performed in the first layer of the FCN classifier and the drop probability was set as 0.5 because the feature maps in the higher layers of the whole framework can be more stable and informative. The mini-batch of SGD was set to eight images. Under the framework of Torch 7 (Collobert, Kavukcuoglu, and Farabet 2011) and CuDNN, the whole process was run on a SuperMicro Workstation with an NVIDIA Tesla K20 for GPU acceleration.

In the experiments, 80% of the total samples were randomized chosen for training and the rest were chosen for testing, and overall accuracy (OA) and kappa coefficient κ are used to evaluate the performance of scene classification methods (Congalton and Green 2008).

3.2. *Performance analysis*

Table 2 presents the performance comparison with other studies, where the first three common DL models that use simple and shallow architectures (≤2 layers) without explicit normalization in the preprocessing obtained OAs of 81.78%, 86.83% and 79.98%. Normalized by linear mapping to [0,1], the DeepSat framework of Basu et al. is based on a small two-layer deep belief network architecture trained by empirically

Table 2. Performances of different methods using the SAT-4 data set.

	Training OA (%)	Testing OA (%)	κ	Training time (s)
DBN	–	81.78	–	–
CNN	–	86.83	–	–
SDAE	–	79.98	–	–
DeepSat	–	97.95	–	–
MLP (*z*-score)	97.10	94.76	0.928	1080.91 ± 73.12
TradCNN (*z*-score)	98.39	98.43	0.979	3353.01 ± 395.51
SatCNN (linear)	99.52 ± 0.01	99.43 ± 0.12	0.992 ± 0.00	4362.69 ± 194.08
SatCNN (*z*-score)	99.56 ± 0.07	99.65 ± 0.04	0.995 ± 0.00	1959.76 ± 216.27

DBN: Deep belief network; SDAE: Stacked Denoising Autoencoder; CNN: convolutional neural network; MLP: multilayer perceptron.

Table 3. Performances of different methods using the SAT-6 data set.

Model	Training OA (%)	Testing OA (%)	κ	Training time (s)
DBN	–	76.47	–	–
CNN	–	79.10	–	–
SDAE	–	78.43	–	–
DeepSat	–	93.92	–	–
MLP (*z*-score)	98.55	97.46	0.966	911.12 ± 27.63
TradCNN (*z*-score)	98.34	98.34	0.978	2763.59 ± 52.46
SatCNN (linear)	99.50 ± 0.13	99.50 ± 0.08	0.993 ± 0.00	2387.26 ± 185.37
SatCNN (*z*-score)	99.57 ± 0.02	99.54 ± 0.07	0.994 ± 0.00	2447.92 ± 63.63

DBN: Deep belief network; SDAE: Stacked Denoising Autoencoder; CNN: convolutional neural network; MLP: multilayer perceptron.

selected DeepSat features, and is the published state-of-the-art for this SAT-data set. Preprocessed with *z*-score, the MLP obtained an OA 94.76% and a κ 0.928, and CNN with a traditional and similar architecture to the best CNN model in Basu et al. (2015), achieved an OA of 98.43% and a κ 0.979. The proposed SatCNN model yielded the best performance in 1959.76 ± 216.27 s, an OA of 99.65 ± 0.04% and a κ 0.995. With an identical SatCNN model, the model trained by linearly mapped input HSR-RS images achieved an OA of 99.43 ± 0.12% and a κ 0.992.

Table 3 compares the proposed approach with the published results on SAT-6 data set. The common DL models obtained OAs of 76.47%, 79.10% and 78.43%, respectively. It was found that *z*-score normalization on SAT-6 was essential because the MLP achieved an OA of 97.46% and a κ 0.966, which achieved higher than the best performance of the DeepSat model (Basu et al. 2015), an OA of 93.92%. That is because either linear normalization or *z*-score normalization is used for all scene patches in SAT-6, which can guarantee the range and variation of input scene patches so that the distribution of each scene patch is standardized. The traditional CNN model also obtained an OA of 98.34% and a κ 0.978; the SatCNN model with linear normalization achieved an OA of 99.50 ± 0.08% and a κ 0.993. When trained with *z*-score normalization, SatCNN yielded an OA of 99.54 ± 0.07% and a κ 0.994, and spent 2447.92 ± 63.63 s.

Experimental results on both data sets infer that the SatCNN architecture is a robust and efficient deep CNN model, while the difference between linear normalization and *z*-score is small when they are applied to smaller HSR-RS scene data sets.

4. Parameter analysis

In this section, we investigate the influence of hyper-parameters in SatCNN model during SGD, including the learning rate α, the momentum λ and the testing ratio of the data set, where α controls the search step of SGD and λ gives additional small steps along the search direction to help model converge faster. The SAT-6 was chosen as the data source for the parameter analysis.

Figure 3(*a*) reports the influence of α, which clearly shows that too low or too high α can lead to a decrease in performance, and a slightly higher learning rate can effectively reduce the training time. When α was set to 0.01, the SatCNN model achieved the best performance within the least training time.

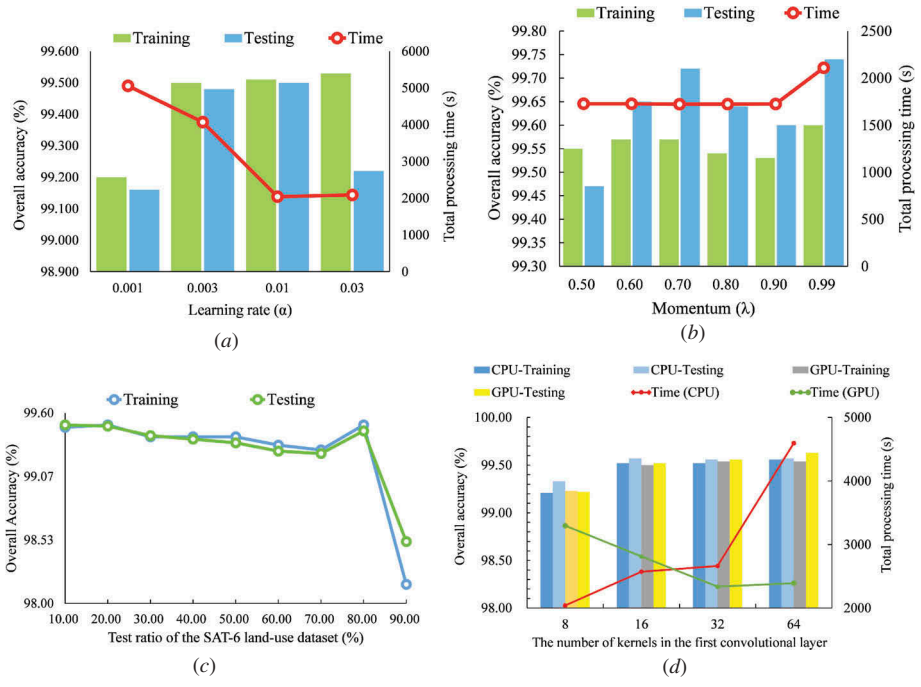


Figure 3. The influence of hyper-parameters, the testing ratios, and the number of kernels in the first convolutional layer on overall accuracy and processing time: (*a*) learning rate *α*, (*b*) momentum value *λ*, (*c*) testing ratio, (*d*) number of kernels in the first convolutional layer.

Figure 3(*b*) presents a comparison of different values of *λ*. If *λ* is not too small, the SatCNN model can converge to a stable status in around 1700 s. Although the final *λ* obtained a better performance, it took more time to perform the additional training epoch so that a *λ* around 0.7 can be considered to be effective in training.

In Figure 3(*c*), the performances of SatCNN with different testing ratios are presented. As the amount of image samples for testing was increased from 10% to 80%, the results of the SatCNN models showed a slight decreasing trend, but still stayed above an OA of 99%. When the testing ratio was set to 90%, the performance of SatCNN quickly dropped below 99% because of the inter- and intra-class complexity of the HSR-RS image. Therefore, the SatCNN model is a relatively effective and robust approach for HSR-RS scene classification.

The efficiency of the GPU acceleration in training the SatCNN models is compared in Figure 3(*d*). The number of feature maps in the first convolutional layer plays an important role in training efficiency: when this value was 8 or 16, training just on the CPU was faster than using GPU acceleration because the numerous times of data transfer between host and device consumed additional time; when this value was 32 or 64, the GPU acceleration started to show its superiority. Furthermore, the last three results trained on the CPU did not show too much difference in testing accuracy but identical architectures showed a slight increasing trend on the GPU-trained models.

5. Conclusion

In this paper, SatCNN – an agile CNN model – has been proposed for HSR-RS image scene classification. The proposed model can balance the model generalization ability and training efficiency. Preprocessed with the *z*-score technique, the SatCNN model can more efficiently capture the intrinsic features inside thousands of the HSR-RS images. With a fine-tuned architecture, SatCNN can be trained directly from the HSR-RS scene data itself and such architecture can effectively handle the inter- and intra-class complexity inside HSR-RS scene. Supported by NVIDIA, CUDA and the CuDNN library, the convolution operations in the convolutional layers can be greatly accelerated, and the total time spent on training the SatCNN model can be greatly reduced. Experiments performed on both the SAT-4 and SAT-6 data sets confirmed that the proposed SatCNN framework is an effective and robust HSR-RS scene classification framework, and the results of the SatCNN models are currently the state-of-the-art for the SAT-4 and SAT-6 data sets. Since the whole SatCNN model can be trained in about 40 min, SatCNN is more practical for further HSR-RS application or even real-time processing.

Acknowledgements

The authors would like to thank the editor, associate editor and anonymous reviewers for their helpful comments and advice. The authors would also like to thank the developers in the Torch developer community for their guidance and help on GPU acceleration. Last but not least, the authors would like to thank researchers who build up SAT scene data sets in Bay Area Environmental Research Institute/NASA Ames Research Center Ramakrishna R. Nemani, NASA Advanced Supercomputing Division and NASA Ames Research Center: Saikat Basu, Robert DiBiano, Manohar Karki and Supratik Mukhopadhyay, Louisiana State University Sangram Ganguly.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Natural Science Foundation of China under Grant Nos. 41622107 and 41371344, and Natural Science Foundation of Hubei Province, China.

References

Basu, S., S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani. 2015. "DeepSat - A Learning framework for Satellite Imagery." *CoRR*: 1-10. doi:10.1145/2820783.2820816.

Bosch, A., X. Muñoz, and R. Mart. 2007. "Which Is the Best Way to Organize/Classify Images by Content?" *Image and Vision Computing* 25 (6): 778–791. doi:10.1016/j.imavis.2006.07.015.

Castelluccio, M., G. Poggi, C. Sansone, and L. Verdoliva. 2015. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks." *CoRR* arXiv:1508.00092. https://arxiv.org/pdf/1508.00092.pdf

Cheriyadat, A. 2012. "Aerial Scene Recognition Using Efficient Sparse Representation." In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, 11. New York: ACM.

Collobert, R., K. Kavukcuoglu, and C. Farabet. 2011. "Torch7: A Matlab-Like Environment for Machine Learning." In *BigLearn, NIPS Workshop*. Curran Associates.

Congalton, R. G., and K. Green. 2008. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. Boca Raton, FL: CRC Press.

Hare, J., P. Lewis, P. Enser, and C. Sandom. 2006. "Mind the Gap: Another Look at the Problem of the Semantic Gap in Image Retrieval." In *Electronic Imaging 2006*, 607309–607309. Bellingham, WA: International Society for Optics and Photonics.

Hu, F., G. Xia, J. Hu, and L. Zhang. 2015. "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery." *Remote Sensing* 7 (11): 14680–14707. doi:10.3390/rs71114680.

Krizhevsky, A., I. Sutskever, and G. Hinton. 2012. "Imagenet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems*, 1097–1105. Curran Associates.

Lazebnik, S., C. Schmid, and J. Ponce. 2006. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories." In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 22169–22178. Piscataway, NJ: IEEE.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444. doi:10.1038/nature14539.

LeCun, Y., L. Bottou, G. Orr, and K. Müller. 2012. "Efficient Backprop." In *Neural Networks: Tricks of the Trade*, 9–48. Heidelberg: Springer.

Simonyan, K., and A. Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In *Arxiv Preprint Arxiv:1409.1556*. https://arxiv.org/pdf/1409.1556.pdf

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The Journal of Machine Learning Research* 15 (1): 1929–1958.

Zhang, F., B. Du, and L. Zhang. 2015. "Saliency-Guided Unsupervised Feature Learning for Scene Classification." *IEEE Transactions on Geoscience and Remote Sensing* 53 (4): 2175–2184. doi:10.1109/TGRS.2014.2357078.

Zhao, B., Y. Zhong, G.-S. Xia, and L. Zhang. 2016a. "Dirichlet-Derived Multiple Topic Scene Classification Model Fusing Heterogeneous Features for High Spatial Resolution Remote Sensing Imagery." *IEEE Transactions on Geoscience and Remote Sensing* 54 (4): 2108–2123. doi:10.1109/TGRS.2015.2496185.

Zhao, B., Y. Zhong, and L. Zhang. 2013. "Scene Classification via Latent Dirichlet Allocation Using a Hybrid Generative/Discriminative Strategy for High Spatial Resolution Remote Sensing Imagery." *Remote Sensing Letters* 4 (12): 1204–1213. doi:10.1080/2150704X.2013.858843.

Zhao, J., Y. Zhong, H. Shu, and L. Zhang. 2016b. "High-Resolution Image Classification Integrating Spectral-Spatial-Location Cues by Conditional Random Fields." *IEEE Transactions on Image Processing* 25 (9): 4033–4045. doi:10.1109/TIP.2016.2577886.

Zhao, W., Z. Guo, J. Yue, X. Zhang, and L. Luo. 2015. "On Combining Multiscale Deep Learning Features for the Classification of Hyperspectral Remote Sensing Imagery." *International Journal of Remote Sensing* 36 (13): 3368–3379. doi:10.1080/2150704X.2015.1062157.

Zhong, Y., Q. Zhu, and L. Zhang. 2015. "Scene Classification Based on the Multifeature Fusion Probabilistic Topic Model for High Spatial Resolution Remote Sensing Imagery." *IEEE Transactions on Geoscience and Remote Sensing* 53 (11): 6207–6222. doi:10.1109/TGRS.2015.2435801.

Zhou, L., Z. Zhou, and D. Hu. 2013. "Scene Classification Using a Multi-Resolution Bag-Of-Features Model." *Pattern Recognition* 46 (1): 424–433. doi:10.1016/j.patcog.2012.07.017.

Zhou, W., Z. Shao, C. Diao, and Q. Cheng. 2015. "High-Resolution Remote-Sensing Imagery Retrieval Using Sparse Features by Auto-Encoder." *Remote Sensing Letters* 6 (10): 775–783. doi:10.1080/2150704X.2015.1074756.