第29回日本情報地質学会講演会 講演要旨集 027-028頁 2018年

Pixel- and Object-based Deep Neural Network LULC Classification using Remote Sensing Data and R Software

Hang T. DO*, Venkatesh RAGHAVAN** and Go YONEZAWA***

Graduate School for Creative Cities, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-5858, Japan
Email: *dohanghumg@gmail.com, ** raghavan@media.osaka-cu.jp , *** yonezawa@media.osaka-cu.jp

Key words: LULC, Deep learning, Object-based, R-keras, SGD

1. Introduction

Deep-learning (DL) algorithms, which learn the representative and discriminative features in a hierarchical manner from the data have been applied for RS data analysis, including Land Use/ Land Cover (LULC) classification (Zhang L. et al., 2016). R is the free software environment for statistical computing and graphics which provides Keras package - a high level neural networks API develop which a focus on enabling fast experimentation. In this study, a deep learning algorithm in Keras named Stochastic gradient descent (SGD) optimizer is explored to establish LULC map from multispectral RapidEye imagery in pixel-based and object-based approaches.

2. Methodology

2.1. Data and study area

Remotely sensed data used is 5 meters RapidEye multispectral dataset acquired on September 2014 covering a part of Lao Cai province, Vietnam. Main LULC classes: water, bare land, road, building, paddy field, terrace field and forest will be classified. Reference data is collected from forest map in scale 1:10000 (Ministry of Agriculture & Rural Development, Vietnam). 144569 points of reference data (number point of each LULC classes is selected based on square area of the class) is separated to three sets: training set (80%); validation set (13%) and test set (7%).

2.2. Methodology

2.2.1. Segmentation

Image segmentation technique is a process of grouping similar neighbor pixels to objects. To assess the dynamics of local variance (LV) from an object level to another, Rate of Change of Local Variance (ROC) measurement is used:

$$\mathrm{ROC} = \left[\frac{L-(L-1)}{L-1}\right] * 100$$

Where L is LV value (defined by standard deviation value of pixels inside segment of target level and (L-1) is the value of next lower level. Peaks in the ROC graph show the object levels at which the segments match the types of objects characterized by equal degrees of homogeneity (Draguţ, L. et al., 2010). Segments at threshold 0.03, 0.14, 0.23, 0.26 and 0.45 are used for deep learning classification (Fig 1).

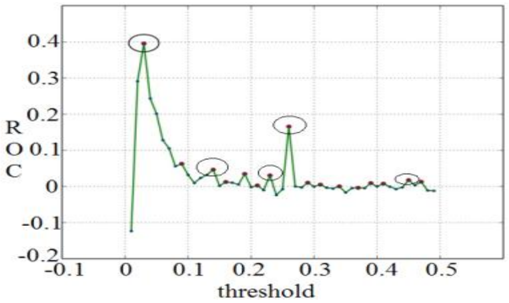


Fig 1. ROC graph, segments at threshold 0.03, 0.14, 0.23, 0.26 and 0.45 are used for next analysis step.

2.2.2. SGD

SGD is a stochastic approximation of the gradient descent optimization and interactive method for minimizing and objective function (Botou, L., 1998). In a simple supervised learning setup, each example z is a pair (x, y) where x is arbitrary input and y is scalar output. The empirical risk $E_n(f)$ measures the performance of training set is calculate:

$$E(f) = \int \ell(f(x), y)\, dP(z) \qquad E_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

Where l is loss function, $f_w(x)$ is the function parameterized by a weight vector w, f is the function which is seek to minimizes the loss $l(f_w(x),y)$, dP(z) is unknown distribution that embodies the Laws of Nature. The expected risk E(f) determines the generalization concert. SGD algorithm is the drastic simplification which is not computing the gradient of $E_n(f_w)$ precisely, each iteration estimates this gradient based on a single randomly selected example z_t:

$$w_{t+1} \;=\; w_t - \gamma_t \nabla_w \, Q(z_t, w_t)$$

The stochastic process $\{w_t,\ t = 1\ \ldots\}$ depends on the selected random examples at each interaction. Since the algorithm does not need to remember which examples were visited through the previous iterations, it can analyze examples on the fly in a deployed system. It leads SGD can optimize the expected risk directly, since the examples are randomly drawn from the reference distribution (Bottou, L., 2010).

2.2.3. SGD parameters setting up

Deep neural network SGD uses five bands of RapidEye image as input of network; output is the seven LULC classes. 5 bands input and reference data are converted to ascii format. Main parameters of the network are established: hidden layers = 4, neurons = 150, l1 = l2 = 0.00001, learning rate = 0.01, activation of input and hidden layers is rectified linear unit ('relu'), of output layer is softmax normalization ('softmax'). The model named "keras_model_sequential", Stochastic gradient descent optimizer is defined with learning rate = 0.01. The model then is compiled using 'sparse_categorical_crossentropy' loss function and training set. Next, the model is fitted to the validation data using callbacks function named "callback_tensorboard". R script is modified based on Github (Zia Ahmed, 2018). Since the training process finishes, test set then uses for assessing accuracy of the classification method. The neural network is shown in Fig 2.

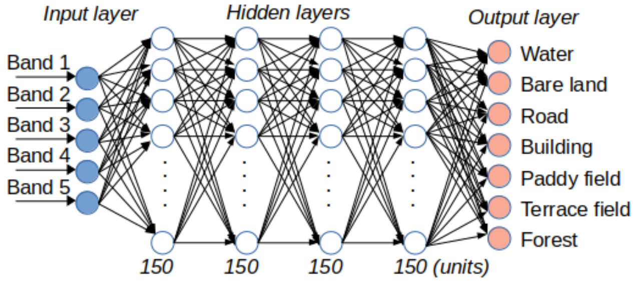
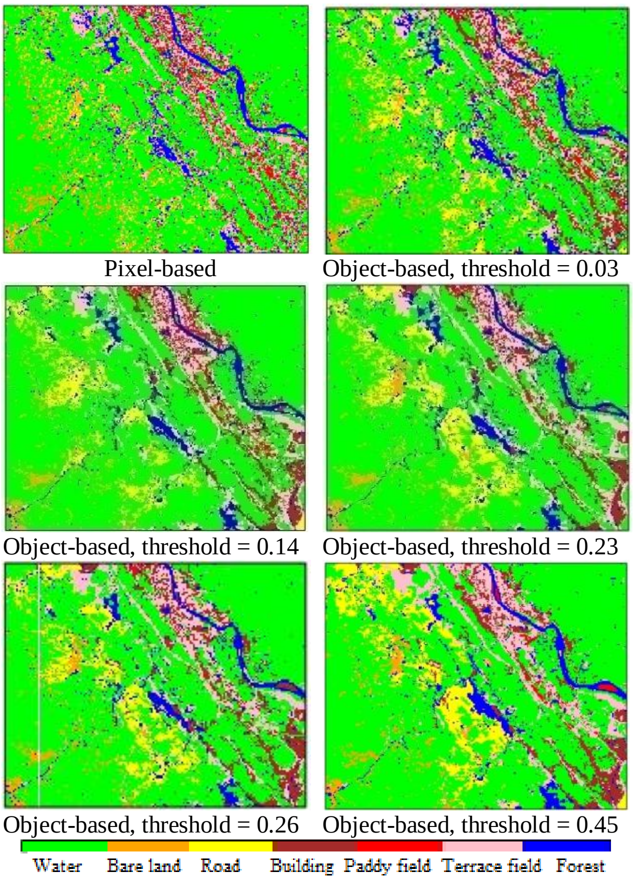


Fig. 2. Deep neuron network model of this study

3. Result and Conclusions

LULC maps in pixel-base and object-base are produced (Fig. 3). Table 1 shows accuracies of six classifier maps. The approaches demonstrate very good classification results since accuracy values are almost equal and range from 94.5% to 94.9%. However, each method can classify effectively different LULC classes. Pixel-based approach illustrates the highest accuracy of water (99.6%), building (90.9%) and paddy field (85.9%), while object-based approaches at threshold=0.03, 0.14, 0.23, 0.26 and 0.45 confirm the best outcomes of terrace (97.1%), forest (97.7%), road (84.1%), bare land (91.2%) and forest (97.7%) correspondingly. Especially, road shows the lowest accuracy value in most of approach cases in comparison with other classes. Road, building and terrace field classes accuracies various in larger range than other classes.

In future work, combination of pixel-based and object-based approach may be established as a more effective classification solution. On the other hand, it needed to find a method that can support to select optimal parameters from a lot of available parameters of SGD optimizer.


Pixel-based

Object-based, threshold = 0.03

Object-based, threshold = 0.14

Object-based, threshold = 0.23

Object-based, threshold = 0.26

Object-based, threshold = 0.45

Fig 3. LULC classification maps

Table 1. Accuracy of Pixel-based and Object-based LULC classification (%)

	Threshold	Water	Bare land	Road	Building	Paddy field	Terrace field	Forest	Overall
Object-based	0.03	99	90.9	81.3	*86.7*	*82.9*	**97.1**	97.6	94.6
	0.14	99.4	89.7	78.4	89.6	84	*95.7*	**97.7**	94.8
	0.23	*98.4*	90	**84.1**	88.4	83.9	96	*97.4*	94.6
	0.26	99.2	**91.2**	83.4	87.3	84.8	96.9	97.5	94.8
	0.45	99.5	89.6	*77.1*	90.1	83.5	96.5	**97.7**	94.9
Pixel-based		**99.6**	*88.5*	79.6	**90.9**	**85.9**	96.2	97	94.5

References

Zhang, Li., Zhang, Le., Du, B. (2016), Deep Learning for remote sensing data. IEEE Geosciences and remote sensing magazine, 0274-6638/16.

Bottou L. (1998), Online Algorithms and Stochastic Approximations. Cambridge University Press, ISBN 978-0-521-65263-6.

Bottou, L. (2010), Large-scale machine learning with Stochastic gradient descent. Proceedings of COMPSTAT'2010, Pyhisca-Verlag,178-179p.

Zia A. (2018), Deep Neural Network with keras-R (TensorFlow GPU backend): Satellite-image classification. https://github.com/zia207/Satellite-Images-Classification-with-Keras-R, retrieved on 25th May 2018.

Bauer, R. and Dahlquist, J., (1998), Technical market indicators: analysis and performance. New York: Wiley, 144p.

Dragut, L., Csillik, Tiede, D., Levick, S. R. (2010), ESP: a tool to estimate scale parameter for multiresolution image segmentation of remotely sensed data. International Journal of Geographical Information Science, 24-6, pp. 859-871.