Aerial Scene Classification with Convolutional Neural Networks

Sibo Jia, Huaping Liu, and Fuchun Sun

Department of Computer Science and Technology,
Tsinghua University, Beijing, China
State Key Lab. of Intelligent Technology and Systems, Beijing, China
hpliu@tsinghua.edu.cn

Abstract. A robust satellite image classification is the fundamental step for aerial image understanding. However current methods with hand-crafted features and conventional classifiers have limited performance. In this paper we introduced convolutional neural network (CNN) method into this problem. Two approaches, including using conventional classifier with CNN features and direct classification with trained CNN models, are investigated with experiments. Our method achieved 97.4% accuracy on 5-fold cross-validation test of the UCMERCED LULC dataset, which is 8% higher than state-of-the-art methods.

1 Introduction

The satellite image analysis has received great interest from both the academic and industrial communities. However, the classification and understanding of the aerial scenes admits many technical challenges such as the diversified classes and obscure image details. To tackle these problems, many modern machine learning methods have been developed to address the aerial scenes classification. A detailed survey can be found in [5].

On the other hand, some deep learning methods, such as auto-encoder, convolutionary neural networks (CNN) and others, have been extensively studied in image classification, speech recognition and machine learning[1,2,3]. All of the successful applications show that stack generalization plays important roles in the machine intelligence. However, to the best knowledge of the authors, the deep learning method has never been used in the classification of aerial scenes. This motivates us to perform experimental validations on the problem.

In this paper, we perform extensive experiments to show that a well-trained CNN can get very surprisingly high recognition accuracy on public available aerial scene dataset. Currently the best accuracy is about 90%, while our method can achieve accuracy of 97%. The rest of this paper is organized as follows: Section 2 gives a brief introduction about CNN. Section 3 presents the details about the classification and Section 4 shows the experimental results.

2 Brief Introduction on CNN

Convolutional neural network (or CNN) is a widely used model for image and video recognition, which features a feed-forward artificial neural network where

X. Hu et al. (Eds.): ISNN 2015, LNCS 9377, pp. 258–265, 2015.
DOI: 10.1007/978-3-319-25393-0_29

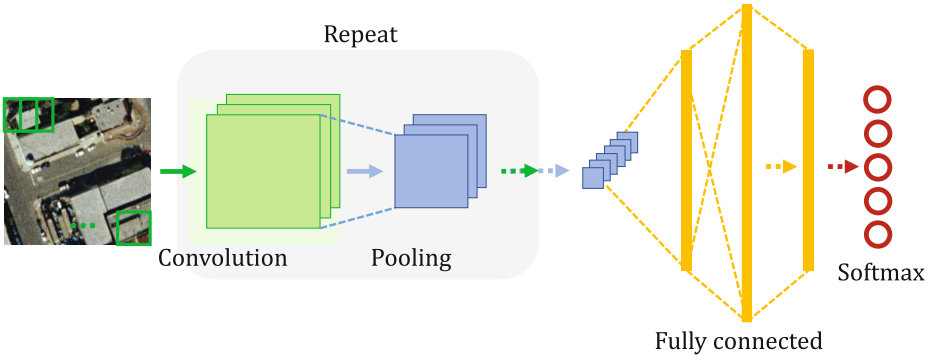

Fig. 1. An exemplary architecture of CNN

the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field. Compared to other image classification algorithms, convolutional neural networks use relatively little pre-processing, as it can learn the filters that in traditional algorithms were hand-engineered. The lack of a dependence on prior-knowledge and the existence of difficult to design hand-engineered features is a major advantage for CNNs.

Figure 1 shows the typical architecture of a CNN network. It consists of multiple layers of small neurons which look at small portions of the input image, called receptive fields. The results of these collections are then tiled so that they overlap to obtain a better representation of the original image. Each neuron consists of a convolution operation with weights W^k and bias b_k and an activation operation $f(\cdot)$. Then the feature of the k-th neuron h^k is obtained by

$$h_{ij}^k = f((W^k * x)_{ij} + b_k)$$

where x is the output feature map of the previous layer. Between the convolutional layers exists local or global pooling layers, which combine the outputs of neuron clusters. When the convolutional and pooling layers are enough to fully cover the whole image region, they are connected to MLP (multilayer perceptron) layers and optionally softmax classification layers. The MLP layers produce a high dimensional vector which can be served as a compact feature of the image, while the softmax layer directly outputs the classification result of the input image. The network is optimized by backpropagation and stochastic gradient descent. It takes a 'mini-batch' of samples each time, compute the gradient $\nabla L(W)$, and obtain the update value V_{t+1} and updated weights W_{t+1} at iteration $t+1$ given the previous weight update V_t and current weights W_t:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t)$$

$$W_{t+1} = W_t + V_{t+1}$$

where the learning rate α is the weight of the negative gradient and momentum μ is the weight of the previous update[4]. Thanks to the computational power of

modern GPU, the network is able to learn from millions of images and achieves outstanding performance on various vision problems.

3 CNN-Based Geographic Image Classification

Previous work on geographic image classification[7,5,6,8] shows that color, texture and local structures are good discriminative features. It turns out that these information can be well captured by a convolutional neural network, thus it's reasonable to believe that the problem of geographic image classification can be tackled with CNN. Furthermore, despite of the difference on image domains, we argue that the CNN model trained on common images can be helpful on our problem, since the size of a typical dataset for CNN training, *e.g.* ImageNet[9] is by far larger than the geographic image dataset we have at hand and the neural network will be able to learn enough discriminative features from common images which are also effective on geographic images.

We propose two approaches of geographic image classification using CNN. The first one is to use a off-the-shelf CNN model to extract high dimensional features of geographic images followed by a traditional classifier *e.g.* SVM. The other approach is to retrain a CNN model using geographic images based on a pretrained model, the process named 'finetuning', and use the new network directly for classification. We will not train a whole new model mainly because we lack the massive amount of training images. While the first method can be very easily applied as it doesn't need any training of neural networks, an adaptation of CNN models trained on common images to the target image domain will hopefully yield better performance. Thus both approaches are investigated in this work.

3.1 Classification Without CNN Retraining

Following the settings of other works, we constrained all the training and testing data to the LULC dataset[5], which contains 2100 land use images of 21 different classes. We used the CNN deep learning framework Caffe as our experiment platform[10], which provides an efficient implementation of deep learning and several off-the-shelf CNN models. The experiment is conducted as follows: high dimensional features of all the 2100 images in the Features of all the images in LULC dataset are extracted with a pretrained model, then part of the images are used to train a classifier while the rest serve as testing data. The training and testing split follows the form of a 5-fold cross validation.

There are three trained models provided by Caffe which we used for our classification problem: AlexNet[2], GoogLeNet[11] and CaffeNet which is an improved version of AlexNet. All three models are trained on the ImageNet dataset, generating features of which dimension ranges from 1024 to 4096. As for the classifiers, we tested SVM, KNN classifier and random forest. As the combinations of model and classifier are rather large, we conduct the experiment in two steps. First we

try different classifiers on one of the trained models, then we use the best classifier setting to test other CNN models. Final result is reported as the average accuracy of cross validation on the best model/classifier combination.

3.2 Classification with CNN Retraining

In this experiment, we will train a CNN model using a trained model and images from the training set. We use CaffeNet as the model to finetune on, which is originally trained on the 1000-class ImageNet images. Instead of using the 4096-dimension features as we did in the previous experiment, this time we will use the softmax classification output. The only modification we make to the CaffeNet is to change the 1000-class softmax layer to a 21-class softmax layer corresponding to the LULC dataset, enabling the network to learn more discriminative features and a 21-class classifier for the LULC dataset. Before training begins, the parameters of every layer except the softmax layer are set to be identical as the trained CaffeNet model, while the softmax layer parameters are initiated randomly. Then the network is trained keeping the learning rate of previous layers smaller than that of the softmax layer, in order to learn the classifier and 'finetune' the convolution layers simultaneously.

We follow the same 5-fold cross validation setting as in the previous experiment. That means only 1680 images can be used to train the CNN model, which is far from enough for a typical deep learning scenario. Thus we extended the training set by flipping and rotating every image to form 7 new images, resulting in a training set 8 times the size to the original. This operation is reasonable for the LULC dataset because content of the photo taken from an aircraft is almost always invariant to flipping and rotation.

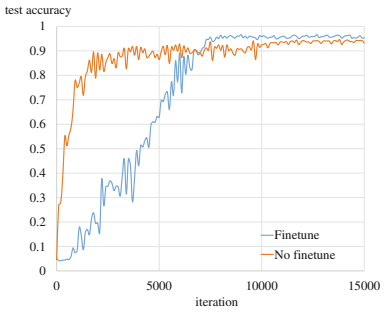

Fig. 2. Training with or without finetune

Due to the small training set, the network only took about 2 hours to convolve on a TITAN BLACK GPU. Testing error after the network convolves is lower than the error rate without retraining CNN. In order to confirm that the improvement is gained from the finetuned CNN instead of from the softmax

classification layer alone, we ran the training process again, keeping everything the same except for fixing the parameters in the convolution layers, which is equivalently training a softmax classifier only. Curves for the training process are shown in Figure 2, which reveal that only training the classifier leads to a faster convolving speed but lower performance. This can be explained by the fact that fewer tunable parameters leads to less learning capacity. Through this experiment, the effect of finetuning the CNN network is also confirmed.

We trained 5 networks in total, each tested on the corresponding 20% testing set and collected the result afterwards. Typically CNN networks are not tested using cross validation, but we did so in order to make a fair comparison.

4 Experiment Results

In this section we report the results of the experiments on the LULC dataset. For every setting accuracies of the 5 cross validation test and average accuracy are reported. First we tested classification on the pre-trained ImageNet CNN features. Accuracy of different classifiers on the same CNN model CaffeNet is shown in Table 1. The best classifier, SVM achieved 94.3% overall accuracy. Fixing the classifier, we tested performance on different CNN models. Table 2 gives the result, showing that the accuracy of CaffeNet is slightly higher than other two models. The experiments show that the CNN network can produce discriminative features good enough to handle the geographic image classification problem, even if the network is not trained on this particular domain.

Table 1. Test result of different classifiers on CaffeNet

Setting	Cross validation accuracy	Overall
SVM	0.94 0.95 0.95 0.95 0.92	**0.943**
KNN Classifier	0.82 0.83 0.85 0.82 0.82	0.829
Random forest	0.89 0.90 0.91 0.88 0.88	0.895

Table 2. Test result of different models on SVM

Model	Cross validation accuracy	Overall
CaffeNet	0.94 0.95 0.95 0.95 0.92	**0.943**
AlexNet	0.93 0.94 0.9 0.95 0.92	0.940
GoogLeNet	0.91 0.93 0.95 0.93 0.91	0.923

Table 3. Test result of classification with new CNN models

Cross validation accuracy	Overall
1.00 0.95 0.96 0.96 0.97	**0.974**

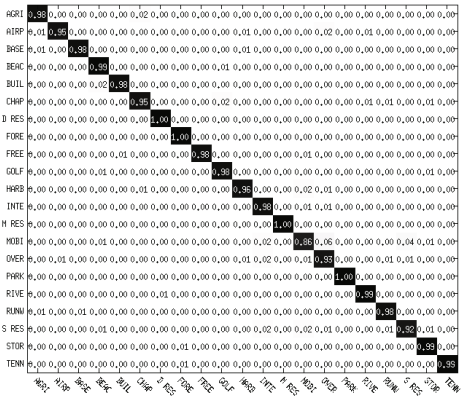

Fig. 3. Confusion matrix of 21 classes

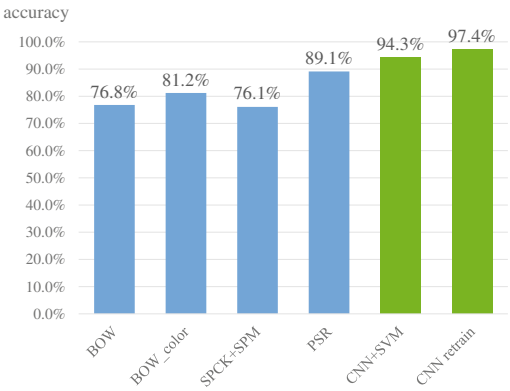

Fig. 4. Comparison with previously reported accuracies

For the finetuned network based on CaffeNet, the result is summarized in Table 3. Overall accuracy of the 5-fold cross validation is 97.4%, when trained on augmented images from part of the LULC dataset. The confusion matrix of the testset is shown in Figure 3. Figure 4 shows the comparison with previously reported accuracies[6]. Time consumption for classifying one image is ∼60ms on an Intel Xeon 2.8GHz CPU.

The statistics of the accuracy for every class is shown in Figure 5, calculated from all the tests of the cross validation. Compared with accuracies of other works, the CNN network is particularly good at capturing textures (*e.g.* chaparral) and structures (*e.g.* intersection), thanks to the learned filters and multi scale pooling.

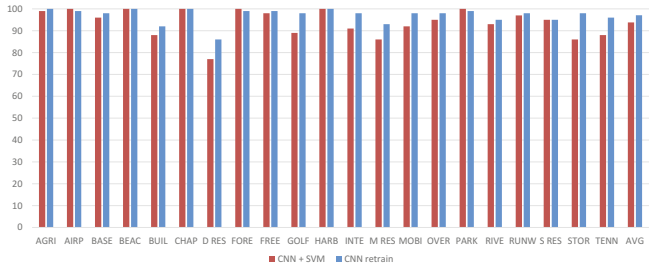

Fig. 5. Accuracy for each class on retrained CNN classification

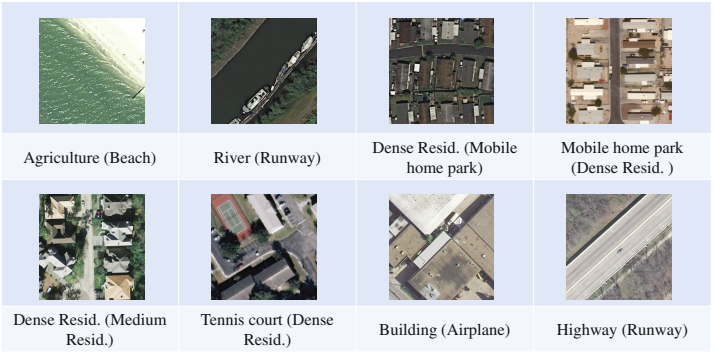

Fig. 6. Some examples of classification error

To explore the limitation and potential improvement, some misclassification samples are shown in Figure 6. Some errors are due to large variation of certain classes, *e.g.* a few patches of 'beach' class are very similar to 'agriculture', however other patches from different angle or scale can never be mistaken as 'agriculture'. This implies that for a practical geographic image classification system, it's necessary to consider neighboring patches to correctly classify hard patches occasionally occurred. One patch of 'tennis court' is classified as residential, as there are indeed many buildings around. This suggests that the current network still needs more training samples or training time to capture particular object like a tennis court. There are also classes containing complicated structures with subtle difference, like 'mobile home park', 'dense residential' and 'building', which might only be better distinguished if given much more training samples.

5 Conclusion

In this work we applied convolutional neural network to aerial image classification problem through two different approaches, and achieved the accuracy of

97.4%, much higher than previous state-of-the-art. Notice that all the training data we used was constrained within the LULC dataset. Analysis of the result showed that the performance may be further improved if given more training data. For future works we plan to extend the problem to aerial scene detection and understanding, and apply state-of-the-art methods of object detection based on CNN, hoping to achieve better performance.

Acknowledgments. This work was supported in part by the National Key Project for Basic Research of China under Grant 2013CB329403; and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20131089295.

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp. 2278–2324 (1998)
2. Krizhevsky, A., Ilya, S., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
3. Hannun, A.Y., Case, C., Casper, J., Catanzaro, B.C., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A.Y.: Deep speech: Scaling up end-to-end speech recognition. In: arXiv:1412.5567
4. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, 2nd edn., LNCS, vol. 7700, pp. 421–436. Springer, Heidelberg (2012)
5. Cheriyadat, A.M.: Unsupervised feature learning for aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 439–451 (2014)
6. Chen, S., Tian, Y.: Pyramid of Spatial Relatons for Scene-Level Land Use Classification. IEEE Transactions on Geoscience and Remote Sensing, 1947–1957 (2015)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169–2178 (2006)
8. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279 (2010)
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. In: arXiv:1409.0575 (2014)
10. Yangqing, J., Evan, S., Jeff, D., Sergey, K., Jonathan, L., Ross, G., Sergio, G., Trevor, D.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678 (2014)
11. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: arXiv:1409.4842