JM-Net and Cluster-SVM for Aerial Scene Classification

Xiaoqiang Lu[1], **Yuan Yuan**[1], **Jie Fang**[1,2]
[1]Center for OPTical IMagery Analysis and Learning (OPTIMAL),
State Key Laboratory of Transient Optics and Photonics,
Xi'an Institute of Optics and Precision Mechanics,
Chinese Academy of Sciences, Xi'an 710119, Shaanxi, China.
[2]University of the Chinese Academy of Sciences, 19A Yuquanlu, Beijing, 100047, China.
y.yuan1.ieee@gmail.com, fangjie713508@gmail.com, luxiaoqiang@opt.ac.cn

Abstract

Aerial scene classification, which is a fundamental problem for remote sensing imagery, can automatically label an aerial image with a specific semantic category. Although deep learning has achieved competitive performance for aerial scene classification, training the conventional neural networks with aerial datasets will easily stick in overfitting. Because the aerial datasets only contain a few hundreds or thousands images, meanwhile the conventional networks usually contain millions of parameters to be trained. To address the problem, a novel convolutional neural network named Justify Mentioned Net (JM-Net) is proposed in this paper, which has different size of convolution kernels in same layer and ignores the fully convolution layer, so it has fewer parameters and can be trained well on aerial datasets. Additionally, Cluster-SVM, a strategy to improve the accuracy and speed up the classification is used in the specific task. Finally, our method surpass the state-of-art result on the challenging AID dataset while cost shorter time and used smaller storage space.

1 Introduction

Aerial scene classification aims to automatically give a semantic label to each image in order to know which category it belongs to, It is a fundamental problem in aerial image comprehending. Recently, aerial image becomes great important for earth observation [Hu *et al.*, 2013; Cheng *et al.*, 2015; Hu *et al.*, 2015; Cheng *et al.*, 2014], Because of the highly complex geometrical structures and spatial patterns, to efficiently understand the semantic information of them is very important, driven by many real-world applications in remote sensing community.

With the development of deep learning, Convolutional Neural Networks (CNNs) achieved state-of-art result on the task of aerial scene classification. However, most of these methods directly use the pre-trained CNN without considering the aerial images characteristics. The pre-trained CNN is only learned on natural images without fine-tuning. Furthermore, it is not feasible to fully train a new CNN model for aerial scene classification [Nogueira *et al.*, 2017]. Because the traditional networks usually contain millions of parameters, therefore, to fully train a new CNN model will require a considerable amount of labeled data and demand high computational costs. Indeed, aerial datasets only have a few hundreds or thousands images. It is difficult to train a high-powered deep CNN with small datasets in practice [Hu *et al.*, 2015]. This paper focus on the following question: *How to fully train a new ConvNet with only few aerial data?* It is a huge demand to design a deep network that are able to effectively train on aerial images.

To address this question, we design Justify Mentioned Network (JM-Net), a novel CNN model to extract the feature vector for aerial scene classification. Compared with traditional Networks [Jia *et al.*, 2014; Szegedy *et al.*, 2015], JM-Net has fewer parameters according to the following steps:

Firstly, to decrease the number of parameters, Justify Mentioned Net (JM-Net) is proposed, which ignores the fully convolution layer. In CNN, the most parameters exist in the fully-connected (FC) layer. In the proposed JM-Net, the FC layers are abandon to decrease the number of parameters.

Secondly, Compress and Expand Convolution module (CEC) is proposed, which has compress layer and expand layer. The CEC module has 3×3 and 1×1 filters in the same expand convolution layer, besides it has fewer 1×1 filters in the same compress convolution filter, so the ECE module helps JM-Net decreases the number of parameters further.

Finally, Cluster-SVM is used to speed up the process of classification. Two images with similar feature vectors belong to the same category, we cluster the feature vectors to some piles, then use SVM to classify the centers of these piles.

In summary, the main contributions of our work include:

1. We have proposed JM-Net, a novel convolutional neural network with one-fortieth parameters of Alex-Net. JM-Net is fully trained on aerial images, achieves good performance on the challenging aerial datasets RSD and AID.

2. We have used Cluster-SVM to speed up the process of classification. Cluster-SVM is a strategy that classify the centers of feature vectors but not the feature vectors themselves.

2 Related Work

Currently, with the development of deep learning, many methods based on it achieve impressive results on many computer vision tasks such as image classification, object recognition, image retrieval, etc. Deep learning methods also achieve state-of-the-art performance on aerial scene classification [Hu *et al.*, 2015], Deep learning methods on aerial scene classification can be divided into two kinds, ***pretrained-CNN method*** and ***retrain-CNN method***.

The pretrained-CNN method is directly using pre-trained deep neural network architectures on the natural images [Russakovsky *et al.*, 2014], the extracted features showed impressive performance on aerial scene classification [Penatti *et al.*, 2015]. The two freely available pre-trained deep Convolutional Neural Network (CNN) architectures are OverFeat [Sermanet *et al.*, 2013] and CaffeNet [Jia *et al.*, 2014]. In [Castelluccio *et al.*, 2015], another promising architecture, i.e. GoogLeNet [Szegedy *et al.*, 2015], was considered and evaluated and this architecture also showed astounding performance. In [Luus *et al.*, 2015], it demonstrated that a multi-scale input strategy for multi-view deep learning can improve the performance of aerial scene classification. Others use the deep-CNN as feature extractor and combine it with feature coding techniques. For instance, Hu et. al [Jgou *et al.*, 2012] extracted multi-scale dense CNN activations from last convolutional layer as features descriptors and further coded them using feature encoding methods like BoVW [Sivic and Zisserman, 2003], Vector of Aggregated Descriptors(VLAD) [Jgou *et al.*, 2012] and Improved Fisher Kernel(IFK) [Perronnin *et al.*, 2010] to generate the final image representation. For all the deep-CNN architectures used above, the features were obtained from the networks pre-trained on natural image datasets and were directly used for classification aerial images.

The retrain-CNN method is training a new deep network with aerial images. However, as reported in [Nogueira *et al.*, 2017], using the exsisting aerial scene datasets to fully train the networks such as CaffeNet [Jia *et al.*, 2014] or GoogleNet [Szegedy *et al.*, 2015] showed a drop in accuracies compared with using the networks as global feature extractors. This because that the large scale networks usually contain a large number of parameters to be trained, therefore, to train them using the aerial datasets with fewer images will easily result in over fitting. Thus, to better fit the dataset, smaller networks for classification were trained [Zhang *et al.*, 2015; Zou *et al.*, 2015]. In [Zhang *et al.*, 2015], a Gradient Boosting Random Convolutional Network(GBRCN) was proposed for classifying aerial images with only two convolutional layers. In [Zou *et al.*, 2015], a deep belief network(DBN) [Hinton *et al.*, 2006] was trained on aerial images, and the feature selection problem was formulated as a feature reconstruction problem in the DBN scheme. By minimizing the reconstruction error over the whole feature set, the features with smaller reconstruction errors can hold more feature intrinsics for image representation. However, the generalization ability of a shallow network is often lower than that of a deep one. It is highly demanded to design a deep network but with fewer parameters.

3 Proposed Method

This section details the proposed method with JM-Net and Cluster-SVM. The flowchart of the proposed is showed in Figure 1.

Firstly, JM-Net is used to extract aerial images' feature vector. JM-Net has the module CEC (including compress layer and expand layer) and ignores the fully convolution layer, so it has fewer parameters and can be trained well on aerial image datasets.

Secondly, cluster the feature vectors' to fewer clusters, and assume each image in the same cluster has the same category label with their cluster center.

Finally, support vector machine is used to classify the cluster centers, then the cluster center' category is used to represent each image's category label in this cluster, and the task of aerial scene classification is finished.

3.1 JM-Net

The overarching of this part is to identify a model that has very few parameters while preserving accuracy on the task of aerial image classification. To address this problem, a sensible approach is to take an existing CNN model (in this paper, we are based on Alex-Net) and compress it in an lossy fashion, and we use the following **4 strategies** to modify the Alex-Net and named it Justify Mentioned Net (JM-Net):

Strategy 1: Inspired by the [Szegedy *et al.*, 2015], which has different size of filters in the same convolution layer, we replace part of the filters from 3×3 to 1×1 in Alex-Net. Given a bugget of a certain number convolutional filters, we will choose to make the majority of these filters 1×1, because that a 1×1 filter has 9X fewer parameters than a 3×3 one.

Strategy 2: Remove the fully convolutional layers, and use average-pooling layer to extract the feature vector from the last convolutional layer. As we all know, the deep convolution layers of CNNs are already have the high-level information and the fully convolutional layers have most parameters of the CNNs. So, to improve the efficiency, we remove the fully convolutional layers.

Strategy 3: Decrease the number of input channels to 3×3 filters. Consider a convolution layer that is comprised entirely of 3×3 filters, The total quantity of parameters in this layer is (number of input channels)*(number of filters)*(3×3). So, to maintain a small total number of parameters in a CNN, it is important not only to decrease the number of 3×3 filters, but also to decrease the number of input channels to the 3×3 filters.

Strategy 4: Postpone the process of downsampling in the network. Downsampling early will led to the information loss early, and decrease the demonstrate capacity of deep layers, and it is proved that delayed downsampling result in higher classification accuracy.

Strategy 1, 2 and 3 are about decreasing the number of parameters in a CNN while attempting to maintain the accuracy, and Strategy 4 is about increasing the accuracy as much as possible.

The CEC-module

Inspired by [Denton *et al.*, 2014], we simply use 1×1 reduction layers followed by the combination of 1×1 and 3×3

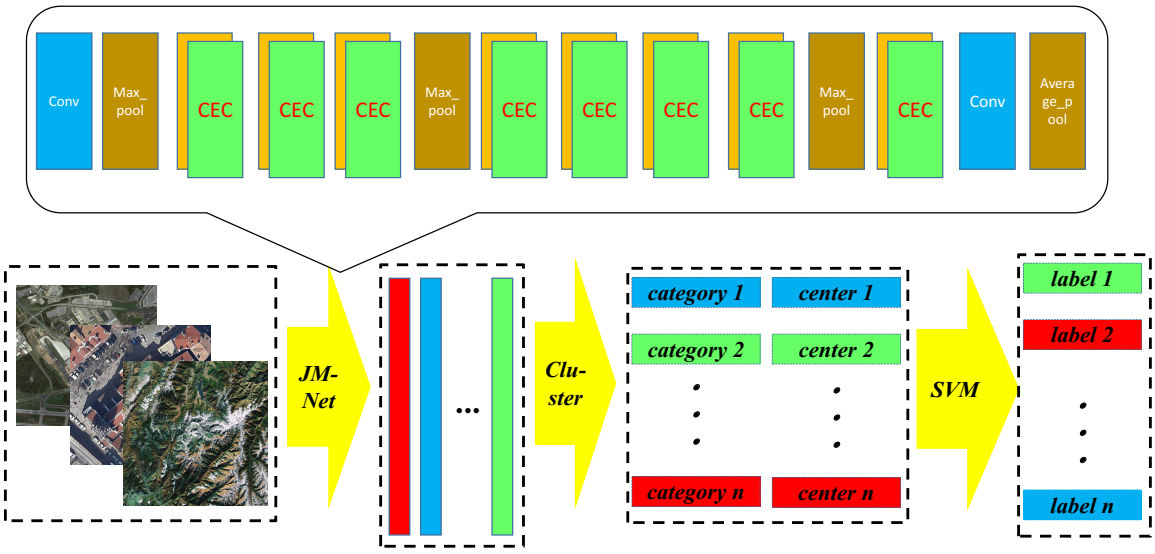

Figure 1: The flowchart of the proposed method, including two important components: JM-Net and Cluster-SVM.

Table 1: The inner architecture of JM-Net

Name_Type	Output_size	Filtersize/Stride	$c_{1\times1}$	$e_{1\times1}$	$e_{3\times3}$
Input_image	$224 \times 224 \times 3$	/	/	/	/
Conv1	$111 \times 111 \times 96$	$7 \times 7/2$	/	/	/
Max_pool	$55 \times 55 \times 96$	$3 \times 3/2$	/	/	/
CEC2	$55 \times 55 \times 128$	/	16	64	64
CEC3	$55 \times 55 \times 128$	/	16	64	64
CEC4	$55 \times 55 \times 256$	/	32	128	128
Max_pool	$27 \times 27 \times 256$	$3 \times 3/2$	/	/	/
CEC5	$27 \times 27 \times 256$	/	32	128	128
CEC6	$27 \times 27 \times 384$	/	48	192	192
CEC7	$27 \times 27 \times 384$	/	48	192	192
CEC8	$27 \times 27 \times 512$	/	64	256	256
Max_pool	$13 \times 13 \times 512$	$3 \times 3/2$	/	/	/
CEC9	$13 \times 13 \times 512$	/	64	256	256
Conv10	$13 \times 13 \times 2048$	/	/	/	/
Average_pooling	2048	/	/	/	/
Softmax	Multi-class	/	/	/	/

convolution filter layers. The architecture of CEC module is showed in Figure 2.

For the sake of simplicity, we define the Compress and Expand Convolution (CEC) module as follows. A CEC module is comprised of: a compress layer which only has $c_{1\times1}$ (compress layer's 1×1 convolution filter), feeding into an expand layer that has both $e_{3\times3}$ (expand layer's 3×3 convolution filter') and $e_{1\times1}$ (expand layer's 1×1 convolution filter) filters. When we use the SF module, we set the number of $c_{1\times1}$ less than the sum of $e_{1\times1}$ and $e_{3\times3}$, so the compress layer helps to limit the number of the input channels to 3×3 filter.

JM-Net Architecture

The JM-Net begins with a convolution layer(conv1), followed by 8 CEC modules(CEC2-9) and a convolution layer, ending with a final Softmax layer. We gradually increase the number of filters per CEC module from the beginning to the end of the JM-Net. the net performs max-pooling with a stride of 2 after conv1, CEC4, CEC8, and conv10; We present the full JM-Net architecture in Table 1.

3.2 Cluster-SVM

Inspired by the super-pixel, we see every image's feature vector as an pixel's RGB feature, through cluster method, we think that the images with similar features belong to the same

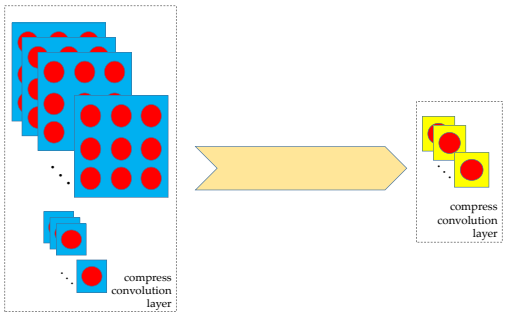

Figure 2: The structure of CEC module, it consists of a compress convolution layer (consists of 1×1 and 3×3 filters) and an expand convolution layer (only consists of 1×1 filter).

category. Among many cluster method, K-means is used widely. However, K-means method gives each element of feature vector the same importance, but there are some problems. For example, if we need to classify a cow and a sheep, the size is more important than the length of fur. In other words, each element of feature has different importance for the classification. In this paper, to improve the classify accuracy, we replace the Euclidean distance with *Sim_loss*, which include both distance and angle information about the feature vector, and the proposed method overcomes the shortcomings described above. The specific correlation function is as follows:

$$Sim\left(x_1, x_2\right) = \lambda \frac{x_1 \cdot x_2}{||x_1|| ||x_2||} + (1-\lambda)\left(1 - \frac{||x_1 - x_2||}{||x_1|| + ||x_2||}\right) \tag{1}$$

The first part of the function is based on consine angle and the second part is based on Euclidean distance, λ is a balanced parameter between 0 and 1, and we make it 0.7 in our experiment, and we replace the Euclidean distance with *Sim_loss*.

Where

$$Sim\_loss\left(x_1, x_2\right) = 2 - Sim\left(x_1, x_2\right) \tag{2}$$

In our work, we use JM-Net to extract features of images, use traditional SVM as classifier when training, and at the testing stage, we use the proposed cluster strategy to speed up the classification process.

4 Experiments

In our experiments, we use two challenging datasets, RSD and AID. RSD consists of 1005 pictures within 19 classes, and AID consists of 10000 pictures within 30 classes (some samples of AID dataset are showed in Figure 3). As we all know, aerial image has the characteristics of scale and direction changeable, so we can not use the images from the dataset only to train a robust Convolutional Neural Network effectively. To address this problem, the datasets are expanded according to the following steps:

1. Randomly extract nine different square regions with different size and direction on each original images, resize all the extracted images to the size of the original images, so we obtain a new dataset which is ten times larger than the original one.
2. Turn all the images obtained in step 1 up and down.
3. Turn all the images obtained in step 2 left and right.

Through above steps, a new dataset with forty times images compare to the original one can be obtained. The schematic is showed as Figure 5.

Figure 3: Samples of AID dataset. AID dataset consists of 10000 images within 30 classes, and the resolution of them are different.

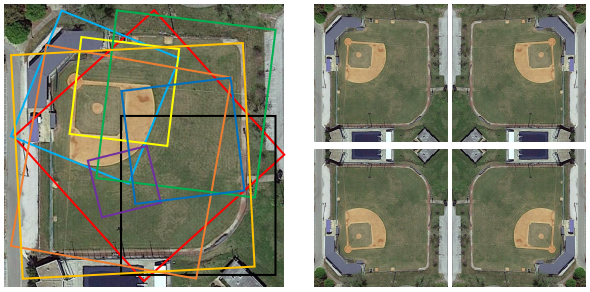

Figure 4: The strategy we used to extend the dataset. The left picture describes the step1, and the right one describes the step 2 and step 3. Through this strategy, the original dataset can be expanded to a new one which has 40 times images of the original dataset.

4.1 Choose the Parameters of JM-Net

According to the proposed method, there are two important parameters needing to choose: The ratio between 3×3 and 1×1 expand convolution filters.

If fewer parameters can be used to demonstrate the feature of an image accurate enough, we can compress the CNN model to some extent. We have used KL method to decrease the dimension of standard VGG-16's and Alex-Net' 4096-D feature vector. In order to simplify the experiment, we decrease the number of feature element by half during each step. Then these features are used to train the classifier. Finally, we calculate classification accuracy as Table 2.

From Table II, it is obvious that 2048 dimension vectors are enough to represent the aerial scene images. when design

Table 3: The influence of the ratio of $c_{3\times3}$ and $c_{1\times1}$ on RSD and AID

Ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
RSD(Acc%)	36.15	50.40	75.36	86.48	**90.30**	90.18	90.05	90.25	90.15
AID(Acc%)	43.56	58.12	64.35	88.63	**89.98**	88.96	89.42	90.04	89.85

Table 2: The Accuracy of different dimension features on RSD & AID

Dim	Acc(VGG16-Net+SVM)		Acc(Alex-Net+SVM)	
	RSDataset	AIDataset	RSDataset	AIDataset
4096	88.16	**88.53**	88.57	87.85
2048	**88.16**	87.47	**88.57**	**87.93**
1024	86.53	86.28	87.35	87.61
512	86.12	85.00	85.72	85.88
256	83.67	82.07	84.90	83.15
128	81.23	76.90	82.45	78.62
64	74.70	69.12	78.78	68.79
32	64.08	56.76	68.16	58.43
16	50.61	39.35	48.98	43.22
8	34.69	23.23	34.69	24.35
4	28.16	17.93	26.12	19.81
2	19.59	13.21	20.82	15.44
1	10.21	7.26	14.69	8.94

the architecture of JM-Net, we make the output of JM-Net's last layer to be 2048 dimensions.

As we all know, a 1×1 filter has 9X fewer parameters than a 3×3 one. In other words, if we want to reduce the parameters of the Convolutional Neural Network, we should use more 1×1 filters and fewer 3×3 filters. When the number of 3×3 filters are reduced, the model perhaps cannot extract typical features of the aerial images and achieve competitive performance on the task. In this section, we will choose the best ratio between $c_{1\times1}$ and $c_{3\times3}$, to use least parameters and remain the classification accuracy. From TABLE 3, It can be seen, when the ratio of 3×3 and 1×1 convolution filters are both 0.5, the classification accuracy reaches to the top.

4.2 Choose the Parameters of Cluster-SVM

For the Cluster-SVM, there are also two important parameters influencing the experiment result: The number of cluster centers and the balance parameter λ of Sim_loss.

Firstly, the number of cluster centers is a very important factor to the final result. Specific to the task of aerial scene classification, if the number of clustering centers is too big, we can not speed up the testing process efficiently. Oppositely, if it is too small, perhaps we will lose the powerful classification capacity of the support vector machine and led to the debasement of classification accuracy. in this section, specific to different datasets, we choose different number of clustering centers to reach the equilibrium. From the TABLE 4 and TABLE 5, we can see when the cluster number is about 1/8 of the amount of the overall images in the datasets, that the classification accuracy reaches the peak.

Additionally, the parameter λ of Sim_loss reflects the attention of the cluster strategy. From the TABLE 6 we can see

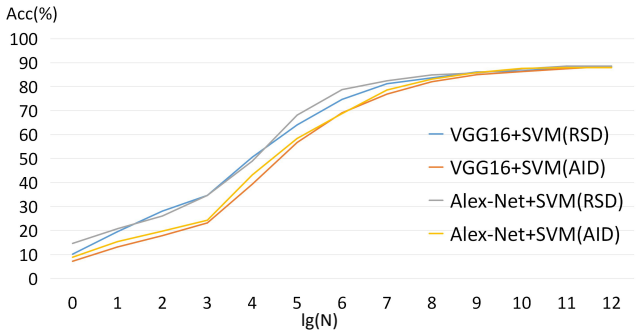

Figure 5: Different vector's dimension result in different classify accuracy, 'N' denotes vector's dimension. From the figure, we can see that when the dimension of feature vector up to a certain level, the classification accuracy doesn't improve further according to the increment of the dimension. The best dimension of the feature vector is about 2048.

that, when the λ is about 0.7, the experiment result reaches to the peak.

4.3 The Performance of Proposed Method

Table 7: The result of proposed method on RSD & AID

Dim	Acc(JM-Net+SVM)		Acc(JM-Net+cluster+SVM)	
	RSDataset	AIDataset	RSDataset	AIDataset
2048	90.20	89.16	**92.20**	**91.91**
1024	89.80	89.01	91.16	90.73
512	90.20	88.12	90.53	89.72
256	87.76	86.92	88.25	87.40
128	85.72	84.75	86.24	85.82
64	83.02	80.15	84.64	83.44
32	80.70	71.12	82.78	74.50
16	71.08	56.76	73.16	61.45
8	52.40	45.36	60.74	50.72
4	40.83	30.23	51.64	34.58
2	25.80	13.21	30.82	27.36
1	13.44	8.26	16.45	13.58

Because JM-Net is modified from the Alex-Net, so we need to compare the performance between these two nets and demonstrate the advantages of JM-Net.

Compare Table 2 and Table 7, Figure **??** and Figure 6. we can see JM-Net gets a better result than Alex-Net, this is because that JM-Net is trained on aerial images and can describe the aerial images better. Besides, From the TABLE VI, we can see from the cluster strategy, the classification accuracy

Table 4: The influence of the number of cluster centers on RSD

Cluster Number	1000	500	250	125	62	31	19
Acc(%)	90.10	90.45	91.50	**92.18**	88.58	67.25	64.37

Table 5: The influence of the number of cluster centers on AID

Cluster Number	10000	5000	2500	1250	625	312	156	78	30
Acc(%)	89.15	89.28	89.56	**90.88**	90.45	89.74	83.27	78.66	65.84

Table 6: The influence of the balance parameter λ on RSD and AID

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
RSD(Acc%)	73.56	75.10	76.84	78.62	81.34	85.68	89.24	**92.25**	90.05	88.10	85.43
AID(Acc%)	68.47	74.35	75.64	74.82	83.54	84.38	88.46	**90.91**	88.25	87.26	83.74

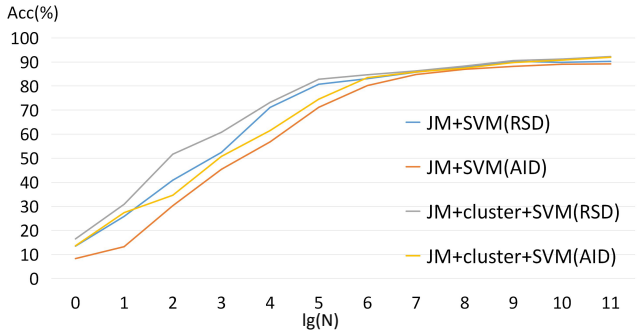

Figure 6: The result of proposed method on RSD and AID datasets. From the figure, we can see that the cluster strategy improves the classification accuracy to a certain degree, especially when the dimension of vector is small.

improves generally, especially when using fewer feature information. The reasons are as follows:

1. Two images with similar feature that belongs to an category, some centers of the feature vector are got through cluster strategy, and these centers are much easier to classify than the raw feature vectors;
2. The cluster center's category is used to represent each image's category label in this cluster. in other words, the cluster strategy decreases the number of testing samples in some terms, and so avoid the likelihood of confusion.

In summary, the proposed method method achieves higher accuracy while using smaller storage space and shorter time.

5 Conclusion

In this paper, we analyzed the difference between natural and aerial images, the dataset of aerial images is much smaller than traditional images, we can not train a standard Alex-Net effectively with aerial images only. For these points, we propose to use a compressed Convolutional Neural Network, JM-Net, which occupies 1/40 space of the Alex-Net and can be stored in portable storage device conveniently, to extract the features of remote sensing images. To speed up the process of the task further, we propose a strategies of clustering to suitable for the real-time applications. Experiment results on datasets RSD and AID demonstrate that the proposed method achieves higher accuracy while used shorter time and smaller storage space, and the proposed JM-Net can be generalized to other aerial image tasks conveniently.

References

[Castelluccio *et al.*, 2015] Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Land use classification in remote sensing images by convolutional neural networks. *Acta Ecologica Sinica*, 28(2):627–635, 2015.

[Cheng *et al.*, 2014] Gong Cheng, Junwei Han, Peicheng Zhou, and Lei Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *Isprs Journal of Photogrammetry and Remote Sensing*, 98(1):119–132, 2014.

[Cheng *et al.*, 2015] Gong Cheng, Junwei Han, Lei Guo, and Zhenbao Liu. Effective and efficient midlevel visual elements-oriented land-use classification using vhr remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):1–12, 2015.

[Denton *et al.*, 2014] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann Lecun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Eprint Arxiv*, pages 1269–1277, 2014.

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

[Hu *et al.*, 2013] Qiong Hu, Wenbin Wu, Tian Xia, Qiangyi Yu, Peng Yang, Zhengguo Li, and Qian Song. Exploring the use of google earth imagery and object-based methods in land use/cover mapping. *Remote Sensing*, 5(11):6026–6042, 2013.

[Hu *et al.*, 2015] Fan Hu, Gui Song Xia, Jingwen Hu, and Liangpei Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, 2015.

[Jgou *et al.*, 2012] H Jgou, F Perronnin, M Douze, J Snchez, P Prez, and C Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

[Luus *et al.*, 2015] F. P. S Luus, B. P Salmon, F Van, den Bergh, and B. T. J Maharaj. Multiview deep learning for land-use classification. *Geoscience and Remote Sensing Letters IEEE*, 12(12):1–5, 2015.

[Nogueira *et al.*, 2017] Keiller Nogueira, Otvio A. B. Penatti, and Jefersson A. Dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539–556, 2017.

[Penatti *et al.*, 2015] Otvio A. B. Penatti, Keiller Nogueira, and Jefersson A. Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? pages 44–51, 2015.

[Perronnin *et al.*, 2010] Florent Perronnin, Jorge Snchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, pages 143–156, 2010.

[Russakovsky *et al.*, 2014] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2014.

[Sermanet *et al.*, 2013] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *Eprint Arxiv*, 2013.

[Sivic and Zisserman, 2003] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, page 1470, 2003.

[Szegedy *et al.*, 2015] C Szegedy, Wei Liu, Yangqing Jia, and P Sermanet. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[Zhang *et al.*, 2015] Fan Zhang, Bo Du, and Liangpei Zhang. Scene classification via a gradient boosting random convolutional network framework. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1–10, 2015.

[Zou *et al.*, 2015] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2321–2325, 2015.