UNSUPERVISED FEATURE CODING ON LOCAL PATCH MANIFOLD FOR SATELLITE IMAGE SCENE CLASSIFICATION

Fan Hu[1,2], *Gui-Song Xia*[1], *Zifeng Wang*[1], *Liangpei Zhang*[1], *Hong Sun*[2]

[1] Key State Laboratory LIESMARS, Wuhan University, Wuhan 430072, China
[2] Electronic Information School, Wuhan University, Wuhan 430072, China

ABSTRACT

This paper presents an improved unsupervised feature learning (UFL) pipeline to discover intrinsic structures of local image patches as well as learn good feature representations automatically for image scenes. In our method, the original image patch vectors embedded in the high-dimensional pixel space are first mapped into a low-dimensional intrinsic space by linear manifold techniques, and then k-means clustering is performed on the patch manifold to learn a dictionary for feature encoding. To generate the feature representation for each local patch, triangle encoding method is applied with the learned dictionary on the same patch manifold. Finally, the holistic scene representations are obtained via the bag-of-visual-words (BOW) framework. We apply the proposed method on an aerial scene dataset. Experiments on the dataset show very promising results and demonstrate that our UFL pipeline can generate very effective local features for image scenes.

Index Terms— Unsupervised feature learning, scene classification, linear manifold, image patch

1. INTRODUCTION

Scene classification of satellite images is a significant task in the intelligent remote sensing field. Effectively encoding the local textural and structural features of image scenes is a practicable way to get low-level features which is the fundamental element of generating holistic feature representation.

Recent years, much work [1–3] has focused on automatically learning good features, alleviating the need for hand-engineered features [4, 5] by utilizing different unsupervised learning algorithms for image classification task. These feature learning systems are collectively called unsupervised feature learning (UFL) methods [1]. UFL methods have the capability of discover low-level structures (e.g. edges) as well as high-level structure (e.g. corners and shapes). Given these powerful low-level and high-level features, images of different categories can be better separated in the supervised classification fashion. However, in general settings of UFL, both training model parameters and feature encoding stage involve large quantities of image patches which are relatively high-dimensional vectors in the raw pixel space and contain great redundancy information. Therefore, typical UFL schemes have extremely high computational cost.

In this paper, we propose an improved version of UFL where we extract features on the image patch manifold. At parameter learning stage, we first apply a linear manifold algorithm to map the training patches into a low-dimensional intrinsic space, and then the model parameter, i.e., the dictionary, is learned by applying K-means clustering on the patch manifold. At feature encoding stage, the local patches sampled from image scenes are identically embedded into the same patch manifold with a linear mapping matrix, and next we extract local features for all the image patch by applying triangle encoding method to the corresponding low-dimensional representation of each one.Extensive experiments on the UCM land use dataset show that we can obtain comparable classification performance with a low computational cost.

2. PATCH MANIFOLD BASED UNSUPERVISED FEATURE LEARNING

2.1. Unsupervised Feature Learning

Here we focus on extracting features from images via unsupervised feature learning method that consist of K-means and triangle encoding. As with the standard UFL pipeline, we firstly perform three steps to learn a dictionary from a set of unlabeled images: (1) extract large quantities of small sub-patches from random locations in unlabeled training images; (2) apply brightness and contrast normalization as well as zero component analysis (ZCA) whitening to the patches as a pre-processing stage; (3) train dictionary D from these patches using K-means clustering as mentioned above.

Note that each patch has dimension r-by-r and has c channels (for natural images, there are only R, G, B channels), so each one r-by-r patch can be represented as a vector in $\mathbb{R}^n$ of pixel intensity values, with $n = r \cdot r \cdot c$. Given the learned dictionary D, a corresponding feature vector ϕ for a new patch x can be achieved by the triangle encoding $\Phi(x; D) : \mathbb{R}^n \mapsto$

This work was partially founded by NSFC project No.91338113.

$\mathbb{R}^K$. The triangle encoding is defined as:

$$\phi_k = \max\{0, mean(d) - d_k\} \tag{1}$$

We are now capable of mapping any r-by-r pixel patch to a K-dimensional feature vector ϕ. For a given image of size m-by-m (with c channels), we then divide the image into a number of square sub-patches of size r-by-r pixels, separated by s pixels each. Rather than learning different dictionaries for each sub-patch of the image, we just simply re-use the same $\Phi(x; D)$ to extract features from each sub-patch. Finally all the feature mapping operation at every location can yield a resulting $((m-r)/s+1)$-by-$((m-r)/s+1)$-by-K dimensional feature representation for the original image. This stage is illustrated in Fig. 1.

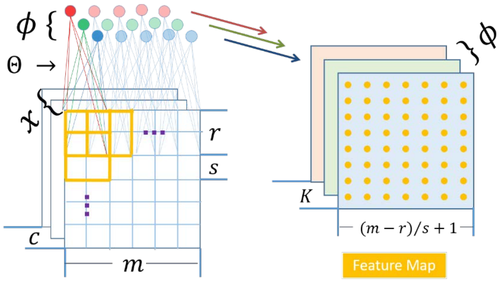

Fig. 1. Illustration of convolutionally extracting local features for images by the UFL approach

2.2. Feature Extraction on Image Patch Manifold

Principal Component Analysis (PCA) is a classical and most popular linear dimensionality reduction technique. However, PCA is incapable of discovering the non-linear structure of the data manifold, thus an alternative linear method called Locality Preserving Projections (LPP) [6] was proposed. Although LPP is a linear algorithms, it shares the locality preserving properties of Laplacian Eigenmaps (LE). In other words LPP gathers advantages from both PCA and LE.

Note that LPP not only has similar properties with LE, but also provide a map for new testing vectors, LPP can be naturally introduced in the UFL pipeline. In dictionary learning stage, the raw image patch vectors (as training vectors) $\{v_i \in \mathbb{R}^n\}_{i=1,\dots,N}$ are first embedded into a low dimensional Euclidean space using a linear manifold technique, then K-means clustering is performed on the low dimensional representation $\{y_i \in \mathbb{R}^d\}_{i=1,\dots,N}$ so as to obtain a dictionary $D \in \mathbb{R}^{d \times K}$ (K is the number of centroids) on the image patch manifold as well as the linear mapping matrix $M \in \mathbb{R}^{n \times d}$. In the feature encoding stage, we generate the low dimensional coordinates $Y \in \mathbb{R}^d$ for new input vectors $X \in \mathbb{R}^d$ by simply multiplying them by the mapping matrix M. Here, these input patch vectors are embedded into the identical low dimensional space which is learned on training patch vectors by LPP. After this, the triangle encoding $\Phi(Y; D) : \mathbb{R}^d \mapsto \mathbb{R}^K$ is applied as a feature extractor to achieve a feature vector ϕ for each input patch vector.

3. SCENE CLASSIFICATION FRAMEWORK

The scene classification flowchart based on our proposed UFL method is depicted in Fig. 2. To summarize, the whole scene classification process can be divided into three separate parts, i.e., local feature extraction, holistic feature representation and SVM classification. Specifically, at the beginning of extracting local feature, we randomly sample a large number of image patches from images in a dataset at any location, then a certain linear manifold method is introduced to yield the low-dimensional representation of the training image patch vectors as well as the linear mapping matrix. Next, we train a dictionary used for feature encoding by applying K-means clustering on the image patch manifold. These few steps are off-line training steps and prepared for the feature encoding step. Following the instructions of Fig. 1, local image patches densely extracted from images on a grid with a fixed step are first projected into the low-dimensional image patch manifold by the linear mapping matrix, and are then encoded into feature vectors via triangle encoding algorithm. After the feature extraction, we train a discriminative codebook according to the improved method described above, and thereby holistic histogram features are generated given the learned codebook. At the last step, we train a SVM classifier and then obtain the category labels for all testing image scenes.

During training and testing stage, the histogram intersection kernel (HIK) [7] for SVM is adopted, which is a very suitable non-linear kernel for measuring similarity between input vectors of histogram type, and is defined as:

$$I(H_i, H_j) = \sum_{k=1} \min(H_i(k), H_j(k)) \tag{2}$$

where H_i and H_j is histogram feature representation for image i and j.

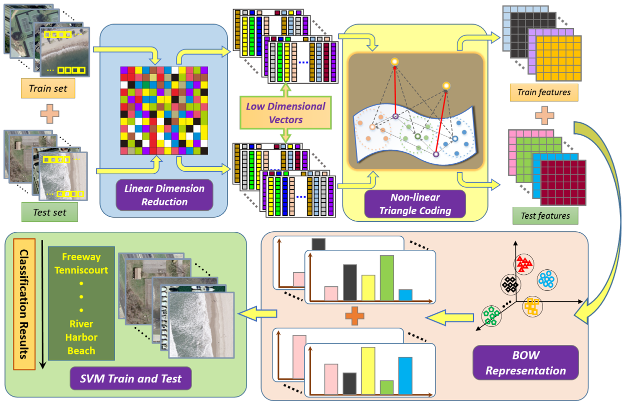

Fig. 2. The proposed overall scene classification framework.

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1. Experimental Dataset and Setup

We evaluate our proposed method on the UCM land use dataset [8], which consists of 21 scene categories which were manually extracted from large aerial orthoimagery with the pixel resolution of one foot. Each class contains 100 images with size of 256*256 pixels.

We randomly samples 100 patches per image and fix the patch size to 10*10 pixels (with R, G, B channels). At feature encoding stage, the sampling step of image patch is set to 5 pixels. We randomly select 80 images per class as training set to train a HIK SVM classifier and the rest as testing set. The classification experiment is repeated 100 times to yield a reliable result. In addition, three important free parameters, which are (i) dimensions d of the low-dimensional space where patch vectors are mapped, (ii) length K of encoded feature vectors, (iii) number of codewords per class L, are varying to investigate how these parameters act on classification results. Otherwise, at the UFL stage, we do ZCA whitening processing on all low-dimensional input samples before applying K-means to learn a dictionary on image patch manifold.

4.2. Experimental Results

We focus our evaluation on three key parameters: d, K and L. While we discuss how the three free parameter influence final classification performance, we evaluate single one (as a variable) of the three and keep the other two to be constants. Now that we extract image patch features on a patch manifold, the dimensions d of low-dimensional space that all image patches are embedded become a dominant role in our experiments. Due to the patch size of 10*10 pixels, thus the length of original patch vectors is 300 and meanwhile limit the size of d within 300. Fig. 3(a) shows the overall scene classification accuracy of UCM dataset under different d ranging from 3 to 200. In this case, K and L is set to 36, 50 respectively by experience. Four typical linear manifold learning algorithms which can learn patch manifold and generate linear mapping matrix is tested here. Random projection (RP) [9] is a simple but effective dimensionality reduction method. It provide the linear mapping matrix Φ where the entries of Φ obey zero-mean, unit-variance normal distribution independently. It is obvious that RP is comparable method to PCA under different d. It is noted that when d is relatively small, say, less than 10, LPP and NPE perform far better than PCA and RP. This is mainly because LPP and NPE have the ability of preserving locality of raw data and loss less information while projecting raw image patches into a low-dimensional space, hence dictionary learning with the corresponding low-dimensional representation seems to be more reliable. In addition, LPP generally performs better than NPE with d varying, especially when d is greater than 30. On the whole, LPP is the most stable and effective one of the four linear manifold methods in our experimental pipeline, and even yield the classification accuracy with 89.2% beyond the performance with original UFL pipeline (learning dictionary and encoding features in the original image patch space).

Table 1. Classification Accuracy Comparison on The UCM Dataset.

Methods	Classification Accuracy (%)
SPM [10]	74
SPCK++ [8]	76.05
SC+Pooling [11]	81.67±1.23
Bag of SIFT	85.37±1.56
Bag of Colors	83.46±1.57
Bag of DisLBP	82.52±2.75
Ours	**90.26±1.51**

Note that triangle encoding method is applied to extract features from local image patches, the quality of local features has a close relationship with the dictionary D. The local features are generated according to (3), hence the number of clusters K when using K-means to train a dictionary D becomes the only parameter that affect the resulting encoded features. Moreover the length of feature vectors is equal to the number of clusters. The length of feature vectors not only directly relates to the computational efficiency, but also has a big impact on the subsequent scene classification performance. Fig. 3(b) shows the overall classification accuracy under different K. In this group of experiments, L and d is fixed at 50, 50 guided by experiments above. Very similar situation to (a), classification performance do not grow consistently along with K in LPP and NPE case. Despite this, LPP still outperform other three linear methods. The improved UFL method with LPP is comparable to the original UFL and performs better when K is less than 36. One possible explanation for this observation is that clustering on patch manifold has more advantages over clustering on the original patch space duo to the natural clustering attribute of LPP when the number of clusters is small.

The average classification performance with varying L which plays an important role in the global feature representations is shown in Fig. 3(c). Once again, we compare our improved UFL method with the original one, in which case the LPP is used and d and K is fixed at 50, 100 respectively. The proposed method performs no worse than the original UFL algorithm, and $L = 100$ produce the best average accuracy of 90.25%. Our results illustrate that too small or too large size of codebook is not beneficial to yield good classification performance, the reason of which is that small size of codebook lack adequate representation ability while large size of codebook lead to high-dimensional global histogram features that may cause overfitting problem during SVM training. In our experiments on UCM dataset, the appropriate L should range

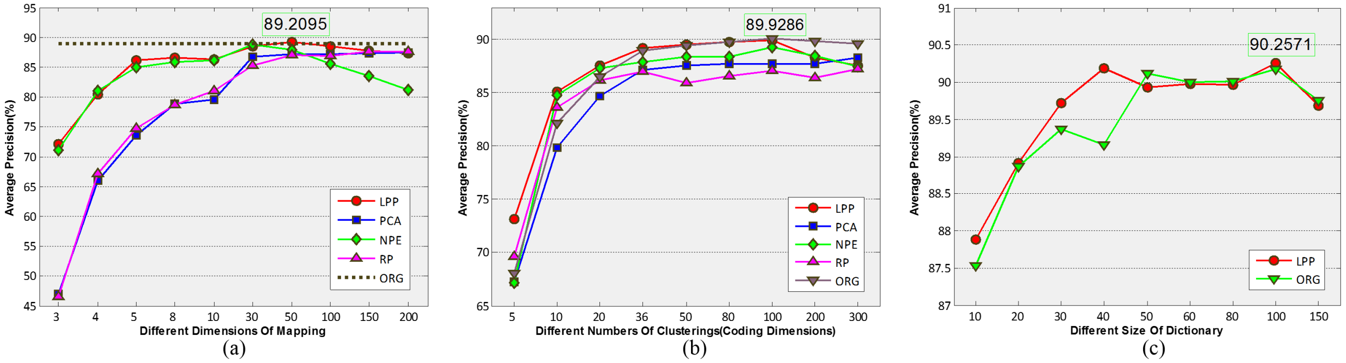

Fig. 3. Classification performance comparison under different parameter settings on UCM dataset. (a) dimensions of low-dimensional space where the patches are embedded; (b) length of encoded features, i.e., the number of clusters at the dictionary learning stage; (c) the size of codebook in BOW representation.

between 40 and 100.

We also compared our proposed UFL method with some off-the-shelf scene classification method that have reported classification accuracy on the UCM dataset. We can see that our method outperform far better than other three classification approaches. In addition, we tested three typical hand-engineered features under the same BOW settings of our method. It is obvious that features automatically learned through our improved UFL pipeline is more suitable than the three well-designed features in scene classification task on UCM dataset. To summarize, our method achieve satisfactory performance with appropriate parameter settings on UCM at a low computational cost.

5. CONCLUSION

This paper presents an improved UFL pipeline, which performs the dictionary learning and feature encoding on the image patch low-dimensional manifold, discovering the intrinsic space of image patches and making dictionary learning and feature encoding more computationally efficient than traditional UFL methods. Experimental results show that the local features generated by the proposed method not only yield better classification performance to some typical hand-designed features, but also are much comparable to the features generated by traditional UFL pipeline with a lower computational cost. Moreover, our scene classification framework based on the simple BOW method and common nonlinear kernel SVM classifier outperforms several off-the-shelf approaches on the UCM dataset.

6. REFERENCES

[1] A. Coates, A.Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *ICAIS*, 2011, pp. 215–223.

[2] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, “Learning invariant features through topographic filter maps,” in *CVPR*. IEEE, 2009, pp. 1605–1612.

[3] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, “Learning mid-level features for recognition,” in *CVPR*. IEEE, 2010, pp. 2559–2566.

[4] G.-S. Xia, J. Delon, and Y. Gousseau, “Shape-based invariant texture indexing,” *Int. J. Comput. Vision*, vol. 88, no. 3, pp. 382–403, 2010.

[5] G.-S. Xia, W. Yang, J. Delon, and Y. Gousseau, “Structural high-resolution satellite image indexing,” in *ISPRS TC VII Symposium-100 Years ISPRS*, 2010, vol. 38, pp. 298–303.

[6] X. He and P. Niyogi, “Locality preserving projections,” in *NIPS*, 2003, vol. 16, pp. 234–241.

[7] S. Maji, A.C. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in *CVPR*. IEEE, 2008, pp. 1–8.

[8] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 270–279.

[9] L. Liu and P. Fieguth, “Texture classification from random features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 574–586, 2012.

[10] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR*. IEEE, 2006, vol. 2, pp. 2169–2178.

[11] A.M. Cheriyadat, “Unsupervised feature learning for aerial scene classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 439–451, Jan 2014.