Pattern Recognition 82 (2018) 118–129

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Improved spatial pyramid matching for scene recognition

Lin Xie[a,1], Feifei Lee[a,1,*], Li Liu[b], Zhong Yin[a], Yan Yan[a], Weidong Wang[a], Junjie Zhao[a], Qiu Chen[c,*]

[a] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China
[b] School of Information Engineering, Nanchang University, China
[c] Major of Electrical Engineering and Electronics, Graduate School, Kogakuin University, Japan

ARTICLE INFO

Article history:
Received 19 May 2017
Revised 20 February 2018
Accepted 26 April 2018
Available online 27 April 2018

Keywords:
Spatial pyramid matching (SPM)
Spatial partition
Histogram of oriented gradients (HOG)
Autoencoder
Scene recognition

ABSTRACT

A scene image is typically composed of successive background contexts and objects with regular shapes. To acquire such spatial information, we propose a new type of spatial partitioning scheme and a modified pyramid matching kernel based on spatial pyramid matching (SPM). A dense histogram of oriented gradients (HOG) is used as a low-level visual descriptor. Furthermore, inspired by the expressive coding ability of autoencoders, we also propose another approach that encodes local descriptors into mid-level features using various autoencoders. The learned mid-level features are encouraged to be sparse, robust and contractive. Then, modified spatial pyramid pooling and local normalization of the mid-level features facilitate the generation of high-level image signatures for scene classification. Comprehensive experimental results on publicly available scene datasets demonstrate the effectiveness of our methods.

1. Introduction

As one of the most challenging problems in the field of computer vision, scene recognition has received considerable attention due to the rapid development of intelligent machines. The approach of placing low-level descriptors (e.g., colour histogram, Local Binary Pattern, and Scale-Invariant Feature Transform) into a classifier directly has been shown to perform poorly [1] because scene images often contain many objects of interest under various backgrounds. Moreover, low-level descriptors are mainly dependent on edges or corner points, and they cannot provide adequate semantic information for scene recognition. Therefore, many researchers have focused on identifying the intermediate semantic representations to narrow the gap between computers and humans with respect to understanding scenes.

Many researchers have attempted to transform low-level descriptors into richer intermediate representations to improve recognition performance and help computers understand more abstract concepts. One extremely popular method is the Bag-of-Visual-Words (BoVW) [2], which is derived from text analysis. BoVW usually involves the following steps. First, local visual descriptors are extracted from image patches, and then dictionary learning produces the codebook, which includes representative visual words. Finally, the image can be characterized by the frequency histogram of the visual words. BoVW discards the spatial structure information in scene images, which restricts the power of the image representations. To overcome this problem, spatial pyramid matching (SPM) [3] based on the BoVW was proposed as a method of incorporating the spatial information of local visual descriptors into the histograms, and it has achieved significant success.

In this paper, the SPM method is used to identify generic spatial structure information within scene images and learn mid-level features. We adopt the histogram of oriented gradients (HOG) as the underlying descriptor because HOG descriptors can be easily and rapidly extracted. To incorporate the generic spatial structure information into the traditional SPM, a new spatial partitioning scheme is proposed to capture a greater degree of local sensitivity in scene images. Partitions in the horizontal and vertical directions are added to preserve consistent structure information. We also modify the pyramid matching kernel to alleviate the influence of viewpoints. This modified SPM achieves better performance and is superior to the conventional SPM in its computational and storage requirements. The steps of this modified SPM are shown in Fig. 1(a). After the K-means clustering on local visual descriptors, the spatial distribution histograms can be calculated. By applying the modified pyramid matching kernel, the histogram representation of the whole image can be obtained. Finally, the intersection kernel SVM (Support Vector Machine) is used to realize the classification.

Another approach named modified spatial pyramid pooling based on various autoencoders is proposed in this paper. Many

* Corresponding authors.
E-mail addresses: feifeilee@ieee.org (F. Lee), q.chen@ieee.org (Q. Chen).
[1] Both authors contributed equally to this work.

https://doi.org/10.1016/j.patcog.2018.04.025
0031-3203/

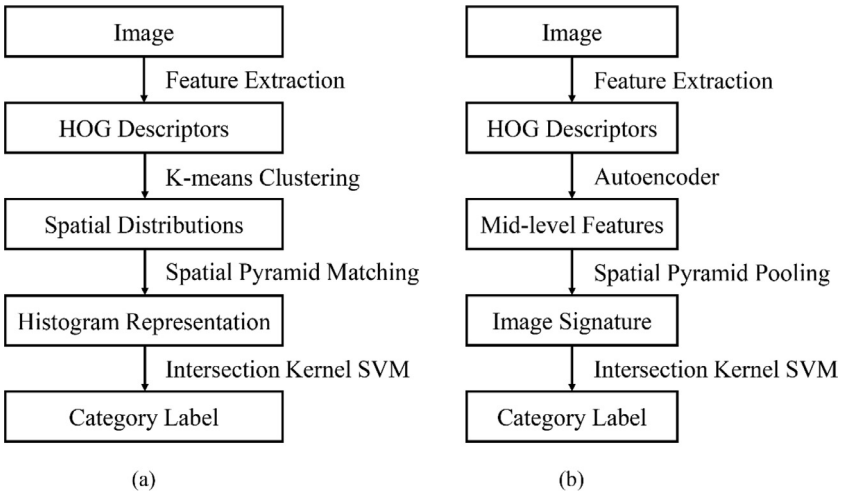

Fig. 1. (a). Major steps of the modified SPM. (b). Major steps of the modified spatial pyramid pooling based on various autoencoders.

models learn representations directly from pixels; in contrast, we explore the encoding of local visual descriptors. As an unsupervised learning technique, the autoencoder is designed to learn an over-complete mid-level feature. A single autoencoder has fewer parameters than other deep architectures, and the directed model facilitates its training. Interesting properties of local visual descriptors are exploited by using three types of autoencoder variants: sparse autoencoder, denoising autoencoder and contractive autoencoder. The learned mid-level features are encouraged to be sparse, robust and contractive. The training process of the autoencoder corresponds to the dictionary learning in the BoVW framework. This method makes the inference of the mid-level features more efficient. Then, the modified spatial pyramid pooling and local normalization on the mid-level features map produce the high-level image signature for scene recognition. This architecture merges the complementary strength of the BoVW framework and autoencoders. Compared with that of other unsupervised means, such as sparse coding, the inference of this model is simple and fast. The main steps of this approach mentioned above are shown in Fig. 1(b).

The remainder of this paper is organized as follows. We review related works in Section 2. The basic techniques are introduced in Section 3. The details of our proposed methods are described in Section 4, including our modified SPM based on HOG, the modified spatial pyramid pooling based on various autoencoders and intersection kernel SVM for scene classification. The experimental results and a discussions are provided in Section 5. Finally, we conclude this paper and offer the suggestions for future work in Section 6.

2. Prior work

Due to the limited power of local visual descriptors, many global features for scene recognition, including GIST [4], CENTRIST [5] and LDBP [6], have been proposed to describe the holistic appearance of a scene. Yu et al. [7] used a unified low-dimensional subspace to effectively fuse multiple features for scene recognition. Scene images can be described as a set of meaningful visual attributes [8,9], although defining of these visual attributes requires significant manual effort and the performance is restricted.

In addition to these elaborate global features, other scene representations are obtained by the popular BoVW model. Many improved versions have emerged over the past few years. For example, Zhou et al. [10] presented a novel Gaussianised vector representation using a global Gaussian Mixture Model (GMM). Qin and Yung [11] extended the BoVW model by introducing contextual information that provides useful cues about the region of interest. Zhou et al. [12] incorporated a multi-resolution representation into the BoVW model. Hotta proposed the local autocorrelation (LAC) feature [13] and local co-occurrence feature [14] based on the subspaces obtained by a Kernel Principal Component Analysis (KPCA) of visual words. Similar to the efforts in the framework of the BoF model, our modified SPM is designed to construct a more discriminative spatial pyramid representation by combining the histograms of visual words from different regions.

In the BoVW framework, many representative visual words constitute the dictionary, feature encoding is used to map the local features to richer intermediate features describing the weights of visual words. Many methods of dictionary learning and feature encoding have been proposed to improve the discrimination of visual words. Wu et al. [15] presented a modified K-means algorithm by incorporating the histogram kernel, and their approach generated a better dictionary because local descriptors were compared in the histogram intersection kernel space instead of the Euclidean space. Yang et al. [16] proposed an extension of SPM utilizing sparse coding of local visual descriptors. Gao et al. [17] introduced the Laplacian matrix in sparse coding to address the problem in which similar local visual descriptors are transformed into different codes. Locality-constrained Linear Coding (LLC) [18] leads to local smooth sparsity and better reconstruction. In our study, we attempt to encode local visual descriptors by autoencoder variants to accelerate the coding process.

In addition to the methods of feature encoding, some researchers have also focused on spatial structure information. For example, Harada et al. [19] estimated the optimal weights of each cell for the most discriminative power, and Jiang et al. [20] developed two classifiers to select the most discriminative pattern from randomized spatial partition schemes. Compared with these models, the method of determining weights and the spatial partition scheme in the approach proposed here is simpler and more effective.

In recent years, deep learning has rapidly developed because of its promising performance for many problems. Deep architectures allow for the exploitation of the potential semantic information behind inputs and provide benefits for classification tasks. However, many deep learning models, such as the Convolutional Neural Network (CNN), Deep Belief Network (DBN) and Deep Boltzmann Machine (DBM), require a large amount of labelled training samples and the consideration of underfitting and overfitting problems during the training process.

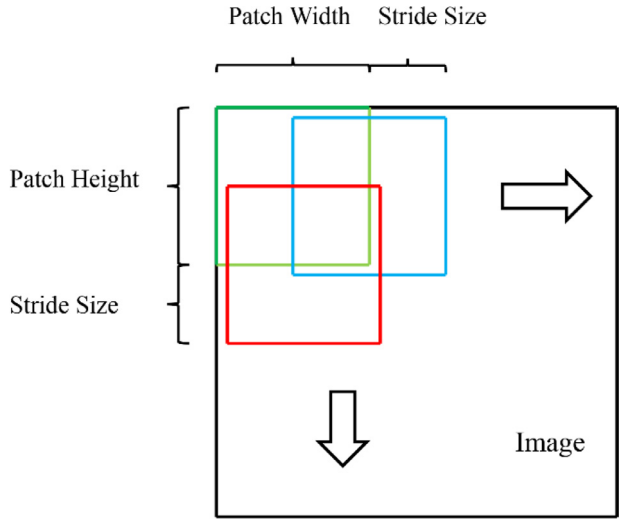

Fig. 2. Patches are densely sampled from images step by step along the vertical and horizontal directions. Subsequently, each patch is decomposed into 2×2 independent cells for extracting HOG descriptors.

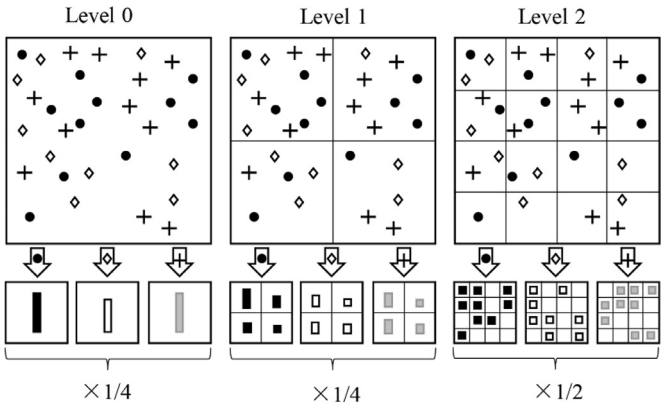

Fig. 3. Toy example of constructing a three-level spatial pyramid. Three types of features are shown in the image. For different spatial regions, we count the number of each type of feature. Finally, the distribution histograms are collected according to the weight.

3. Background

3.1. Local visual descriptors

In our study, local visual descriptors are extracted from patches densely located in the image window as shown in Fig. 2, where these adjacent patches are overlapped. Local visual descriptors calculated from dense regular grids provide for improved scene recognition [21] because this method is capable of capturing the features of uniform regions such as ocean, forest and sky.

In this work, HOG is selected as the local visual descriptor for scene recognition. HOG is a visual descriptor that is widely used in pedestrian and object detection. The primary idea underlying HOG is to characterize object appearance or shape by the distribution of intensity gradients and edge directions in local cells. The descriptors of cells are normalized using a block-wise pattern, which reduces the negative impact of local translations or rotations. This property can meet the requirement of detection of small objects against the large background.

There are two major variants of HOG: the original Dalal–Triggs variant [22] and the UoCTTI variant [23]. The difference between them is that the UoCTTI HOG calculates an extra texture-energy feature and performs a compression on the results in addition to the directed and undirected gradients in the original Dalal–Triggs variant. In our preliminary experiments, better results can be obtained when 2×2 neighbouring HOG descriptors are stacked together to describe a patch. Moreover, the UoCTTI variant appears to improve the recognition performance compared with the original Dalal–Triggs variant; therefore, the stacked UoCTTI HOG descriptors are adopted as the local visual descriptors in the following experiments. To validate the effectiveness of these HOG descriptors, another common visual descriptor, SIFT (Scale-Invariant Feature Transform), is used for comparison purposes in the subsequent section.

3.2. Spatial pyramid matching (SPM)

In the BoF framework, examples are often represented by a set of unordered features with various cardinalities. To precisely match two collections of features, the pyramid match (PM) kernel was introduced in [24]. If pyramids have L levels, then the PM kernel is defined as follows:

$$k(X,Y) = \frac{1}{2^L} I^0 + \sum_{l=1}^{L} \frac{1}{2^{L-l+1}} I^l \tag{1}$$

where I^l represents the matching number of image X and Y at level l. The weight associated with the level is inversely proportional to the grid size at that level to encourage more precise matching at finer geometric levels.

PM allows a weighted histogram intersection to be calculated in multi-resolution histograms; however, spatial information is ignored. Therefore, SPM was proposed to incorporate spatial information into the histogram intersection. SPM conducts PM in the two-dimensional image space by dividing the image into finer-grained parts and calculating the distribution histograms of subregions at different levels of resolution. This spatial pyramid is constructed as shown in Fig. 3. Instead of varying the image resolutions, SPM calculates the distribution histogram at different spatial resolutions from images with the same size. The underlying local descriptors are usually assigned to m discrete types by traditional clustering techniques, such as K-means and the GMM, with each type corresponding to a channel. SPM is performed for each channel. Therefore, the SPM kernel is defined by the sum of matching values of m channels as follows:

$$K(X,Y) = \sum_{c=1}^{m} k(X_c, Y_c) \tag{2}$$

where k denotes the PM kernel and X_c and Y_c are the distribution histograms of the cth feature over all spatial parts. By calculating the occurrence frequency of each type of feature in each part, images with a different number of features can be represented by the distribution histograms with the same dimensionality. Thus, SPM employs the PM kernel to match each type of feature of two images at different spatial resolutions, thereby preserving partial spatial information, which is critical for scene recognition.

3.3. Autoencoder variants

Autoencoder is a special type of neural network composed of two parts: an encoder and a decoder. The encoder aims to learn the hidden representation by effectively encoding its inputs, and the decoder tries to reconstruct the initial inputs from the hidden representations. Autoencoder is often regarded as a technique for dimensionality reduction by restricting the dimension of hidden representations. If the number of hidden nodes is less than that of the visible nodes (i.e., input and output layer), and the linear

activation function and L2 loss function are applied in the autoencoder, then the learned hidden representations correspond to the projection onto the subspace generated by the principal components of inputs [25]. More useful and interesting properties can be obtained when non-linear activation functions are employed in the autoencoder [26]. The basic autoencoder and three types of variants are presented in the following passages.

The encoder is equivalent to a function f that maps the input $x \in \mathbb{R}^{d_x}$ to the hidden representation $h(x) \in \mathbb{R}^{d_h}$. The formulation of encoder is as follows:

$$h = f(x) = s_f(W_1 x + b_h) \tag{3}$$

where s_f is a nonlinear activation function parametrized by a weight matrix $W_1 \in \mathbb{R}^{d_h \times d_x}$ and a bias vector $b_h \in \mathbb{R}^{d_h}$. Similarly, the decoder maps hidden representation h back to reconstruction y with the following form:

$$y = g(h) = s_g(W_2 h + b_y) \tag{4}$$

where s_g is the activation function of the decoder. The parameters of the decoder are the weight matrix $W_2 \in \mathbb{R}^{d_y \times d_h}$ and bias vector $b_y \in \mathbb{R}^{d_y}$. The alternative activation functions are typically the identity function, sigmoid function, hyperbolic tangent function and rectified linear function.

The purpose of training an autoencoder is to identify appropriate parameters $\theta = \{W_1, b_h, W_2, b_y\}$ that minimize the errors between the inputs and reconstructions on a training set D_n. The corresponding objective function is shown as follows:

$$J_{AE}(\theta) = \sum_{x \in D_n} L(x, g(f(x))) \tag{5}$$

where L is the loss function for the inputs and reconstructions. The quadratic loss function: $L(x, y) = \|x - y\|^2$ is generally used to measure the difference between inputs and reconstructions. Cross-entropy loss function is a better choice for the sigmoid activation function [27].

One of the autoencoder variants is the sparse autoencoder, which imposes sparsity on hidden representations. In the sparse autoencoder, the number of hidden nodes is greater than that of the visible nodes. A sparsity constraint is added into the objective function of the autoencoder to learn useful structure information behind the input data. Then, its object function can be written as follows:

$$J_{SAE}(\theta) = \sum_{x \in D_n} L(x, g(f(x))) + \lambda \sum_{x \in D_n} |f(x)| \tag{6}$$

where parameter λ controls the sparsity of hidden representations. Here, the sparse penalty term is the L1-norm of the hidden representations.

To make the hidden representations robust to the partial corruption of inputs, a denoising autoencoder [28] is proposed to recover the original undistorted inputs. At the training stage, the initial input x is corrupted by additive Gauss noise Δx as shown below:

$$\begin{aligned} &\tilde{x} = x + \Delta x \\ &\Delta x \sim \mathrm{N}(0, \sigma) \end{aligned} \tag{7}$$

Then, $\tilde{x}$ is fed into an autoencoder to reconstruct the uncorrupted inputs x. In the test case, the denoising autoencoder directly learns the robust hidden representations from initial input x.

A contractive autoencoder is another variant that also yields robust hidden representations by adding a specific penalty term. This penalty term is defined as the Frobenius norm of the Jacobian matrix of the hidden activations with respect to the inputs. This penalty has been found to help capture the local directions of variation associated with the inputs [29]. The objective function is defined as follows, where $\mathcal{J}_f(x)$ denotes the Jacobian matrix of f with respect to x.

$$\begin{aligned} J_{CAE}(\theta) &= \sum_{x \in D_n} \left(L(x, g(f(x))) + \lambda \left\| \mathcal{J}_f(x) \right\|^2 \right) \\ \left\| \mathcal{J}_f(x) \right\|^2 &= \sum_{ij} \left(\frac{\partial f_j(x)}{\partial x_i} \right)^2 \end{aligned} \tag{8}$$

The power of autoencoders for learning useful features has been confirmed on the handwritten digit database because the contents of handwritten digit images are similar and the size of images is suitable for training autoencoders. However, scene images usually consist of a number of different and unrelated objects, and the large resolution increases the time required to train the autoencoders. Compared with other deep learning models that learn features from pixels, in our model, we try to encode powerful local descriptors into mid-level features using autoencoder variants to overcome these problems and further abstract the low-level features.

4. Proposed algorithms

4.1. Modified spatial pyramid matching

Conventional SPM calculates the distribution of local descriptors within even grids on different levels. This scheme of spatial partitioning generally splits continuous landscapes into fragments, and the local visual descriptors extracted from these fragments are inconsistent and trivial. Conventional SPM has a negative impact on the statistical characteristics of distribution histograms and may result in a bad scene recognition performance. To overcome this problem, the divisions in horizontal and vertical directions are added into the spatial pyramid as shown in Fig. 4 to preserve the spatial structure information, such as continuous shapes or specific layout. This method can be generalized to more than two levels as shown in Fig. 5 if necessary, although a two-level spatial pyramid is adopted here, and a three-level spatial pyramid is used in the traditional SPM. The two-level spatial pyramid is adopted in this paper, because finer divisions lead to some ordinary or similar parts containing a common item or a plain background, and the distribution histograms of these parts from different classes are easily confused in the classification.

The standard spatial partitions and our extra partition schemes of two images from the same category are displayed in Fig. 6. Compared with the image patches in (a) and (c), the spatial regions in (b) and (d) preserves more consistent and continuous information (e.g., the structure of buildings, the vehicle flow and the trends of roads). Although finer partitions can provide more precise feature matches, fragmented information on objects or backgrounds that occur in many other parts may be included, which reduces the ability to discriminate the ultimate image representation.

In addition to the above changes, the weight of each level in the collected histogram is also important for final recognition. The weight of each distribution histogram is dependent on the PM kernel as shown in Eq. (1), which is considered for precise matching of the same objects. However, the appearance of objects and the spatial layout may vary under changes in viewpoint. As shown in Fig. 6, spatial matching is more precise on high levels; however, this matching is not appropriate for a similar scene with different viewpoints. For example, the trends of roads and the positions of cars or buildings are different for the two images in Fig. 6; therefore, the partition in the horizontal direction appears to provide more useful information.

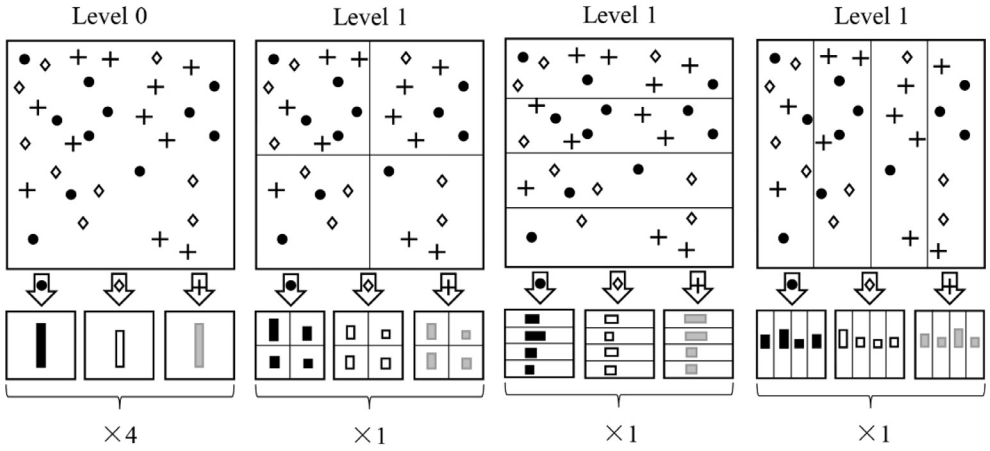

Fig. 4. Toy example of our modified spatial pyramid. Three types of features are included in the image. The divisions in the horizontal and vertical directions are added into level 1. The weights of the distribution histograms depend on our new pyramid matching kernel.

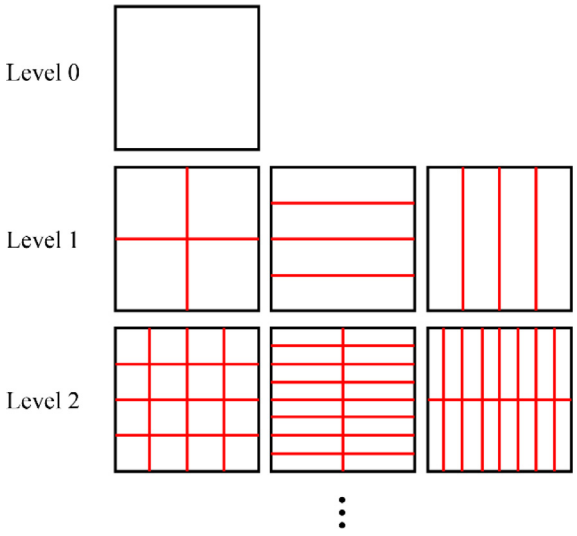

Fig. 5. Our modified spatial pyramid when it is generalized to more than two levels.

According to these traits, we alter the pyramid matching kernel as follows:

$$k(X,Y)=\sum_{l=0}^{L}2^{a(L-l)}I^{l} \tag{9}$$

where a is the kernel parameter. The effects of this parameter on the recognition performance is discussed in Section 5.2.

4.2. Modified spatial pyramid pooling based on various autoencoders

4.2.1. Modified spatial pyramid pooling

An autoencoder is usually employed to learn more compact features and reduce the dimensionality of inputs when the number of hidden nodes is less than the number of visible nodes. The compressed representation requires less storage and has benefits for further computation. In this paper, we seek to learn an overcomplete dictionary from local visual descriptors; therefore, the dimensionality of hidden representations is set to be higher than that of the inputs as shown in Fig. 7.

The autoencoder is used to encode HOG descriptors into richer intermediate representations in the higher dimensional space and effectively decode these hidden representations into the initial

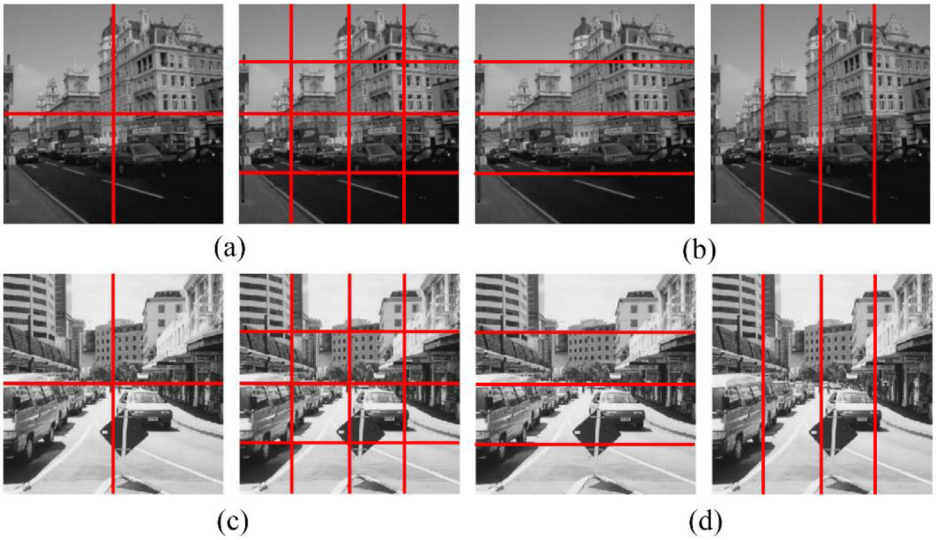

Fig. 6. Spatial partitions of two images from the same category. (a) and (c) Standard partitions in the original SPM; (b) and (d) newly added spatial partitions in our modified SPM.

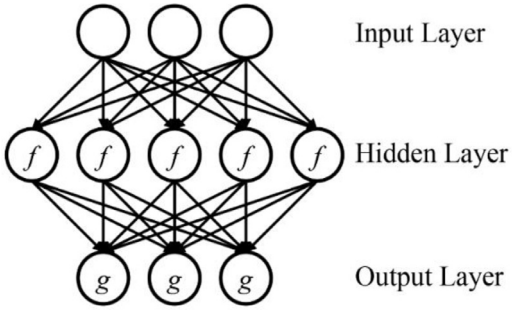

Fig. 7. Structure of the autoencoders used in our study. The notations f and g stand for the activation functions of hidden and output nodes.

inputs. The output of the encoder is regarded as our mid-level features. Different autoencoder variants, which are described in Section 3.3, are used to endow the mid-level features with certain interesting characteristics.

Instead of calculating the distribution histogram of local visual descriptors over each region in the modified spatial pyramid matching algorithm described in Section 4.1, here the modified spatial pyramid pooling and the local normalization is proposed to produce the representations of various regions. Spatial pyramid pooling is performed on the mid-level features map to capture the salient features over each spatial region as shown in Fig. 8. The spatial pyramid is constructed according to our proposed schemes mentioned in Section 4.1. Max pooling has been shown to outperform other available pooling strategies by many algorithms [30]. Given the set of mid-level feature vectors from a spatial region $R = \{h_p \in \mathbb{R}^{d_h} \mid \forall p \in \{1, 2, \ldots, n\}\}$, where n denotes the number of mid-level feature vectors in the spatial region, max pooling aims to aggregate them into a single pooled feature vector $z \in \mathbb{R}^{d_h}$, which can be calculated by the formulation (10), where the subscript q is the element index of the feature vector.

$$z_q = max\left\{h_{1q}, h_{2q}, \ldots, h_{pq}, \ldots, h_{nq}\right\}, \ \forall q \in \{1, 2, \ldots, d_h\} \tag{10}$$

This process produces the pooled features that more robust to local transformations and simultaneously integrates multiple mid-level feature vectors in a spatial region. Moreover, local normalization defined by formulation (11) is applied to the pooled features for compactness and discrimination.

$$u_q = \frac{z_q}{\|z\|_2}, \ \forall q \in \{1, 2, \ldots, d_h\} \tag{11}$$

The $u \in \mathbb{R}^{d_h}$ is the local normalized feature vector, which corresponds to the coloured box as shown in Fig. 8. These local normalized features vectors calculated from all spatial regions are concatenated to form the final high-level image signature.

The autoencoder not only attempts to reserve most visual information from local descriptors, but also plays a critical role in exploiting interesting representations with desirable properties, such as sparsity, robustness and contraction. Therefore, the optimization problem of training the autoencoder is significant, good minimization of the objective function can lead to expected mid-level features. In practice, the evaluation in gradient descent method is expensive due to substantial training samples, so the stochastic gradient descent has been widely used to optimize the objective function, such as convolutional neural networks [31], stacked restricted Boltzmann machines [32] and deep recurrent neural networks [33]. The normal stochastic gradient may result in poor convergence within the limited number of iterations. To obtain better solutions and faster convergence rate, Adam (short for Adaptive Moment Estimation) [34] is considered as the optimization algorithm in the backpropagation, which is computationally efficient with little memory requirements and is proper for problems that are large in terms of parameters. The Adam algorithm learns the appropriate weights and biases of the autoencoder by an iterative process until these parameters converge. To accelerate the convergence, Adam calculates adaptive learning rates for each parameter according to its first and second moment estimations of the gradients.

4.2.2. Modified sparse autoencoder

Among the autoencoder variants mentioned above, the sparse autoencoder appears to have more potential for scene recognition according to our experimental results in Section 5.3. It is necessary to reconsider the sparsity induced by the L1 regularization term and the rectified linear activation function in Eq. (6). On the one hand, the L1 regularization ensures that most hidden outputs can approach to zero, however, for different visual descriptors, the distribution of their active hidden nodes may be similar, which reduces the effective capacity of the sparse representations. To this end, the KL-divergence penalty term is introduced as the

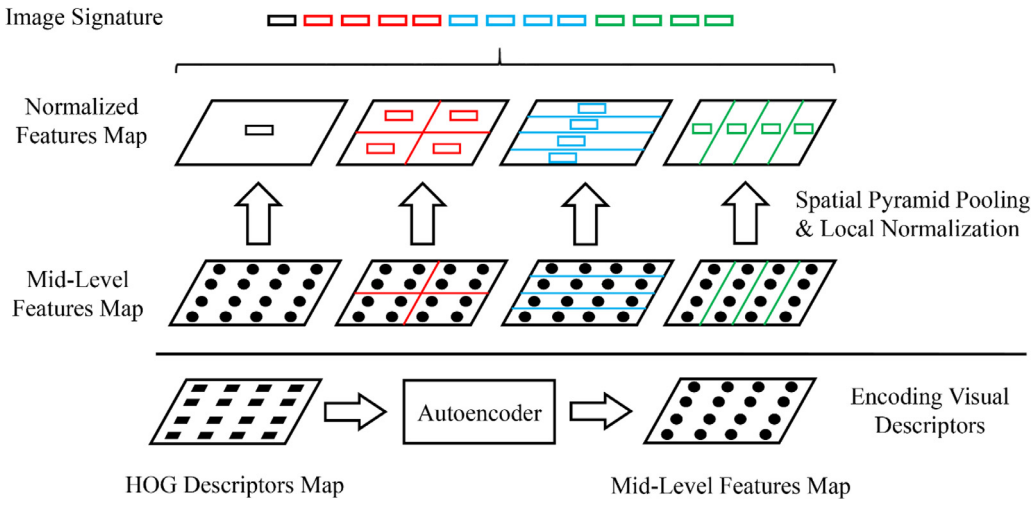

Fig. 8. Process of generating the high-level image signature. HOG descriptors are encoded into mid-level features by an autoencoder. Spatial pyramid pooling is performed on the mid-level feature maps. After normalizing the pooled features, the features are concatenated to form the high-level image signature. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

new sparse regularization:

$$KL\left(\rho \parallel \hat{h}\right) = \sum_{j=1}^{d_h} \rho \log \frac{\rho}{\hat{h}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{h}_j} \tag{12}$$

where ρ is a sparsity parameter, typically a small value close to zero (e.g., $\rho = 0.05$), $\hat{h}_j$ is the average activation of hidden node j, this penalty is equal to zero if $\hat{h}_j = \rho$, and otherwise it increases monotonically as $\hat{h}_j$ diverges from ρ. This sparse regularization ensures that each hidden node can be activated under certain conditions. On the other hand, there is no upper bound for the rectified linear activation function, which may hurt the effectiveness of the sparsity, for example, the reconstruction from the hidden representations to the original inputs relies on few certain active nodes, and the outputs of other active nodes are always close to zero. To avoid this problem, the activation function of hidden nodes is redefined as follows:

$$h_j = \begin{cases} 0, & v_j < 0 \\ v_j, & 0 \le v_j \le 1 \\ 1, & v_j > 1 \end{cases} \tag{13}$$

where v_j denotes the input of the jth hidden node, so the outputs of the hidden nodes range from 0 to 1. In addition, the weight decay regularization is introduced in the objective function to decrease the elements in the weight matrix W_1. All these strategies are aimed to induce a real sparse distribution of the active hidden nodes, so the modified objective function of the sparse autoencoder is presented as follows:

$$J_{SAE'}(\theta) = \sum_{x \epsilon D_n} L(x, g(f(x))) + \lambda KL\left(\rho \parallel \hat{h}\right) + \beta \|W_1\| \tag{14}$$

4.3. Intersection kernel SVM

Let $u^{(t)}$ denote the histogram representation or high-level signature of image $S^{(t)}$. As a binary classifier, a SVM is designed to learn a decision function

$$f(u) = \sum_{t=1}^{N} \alpha_t \kappa\left(u, u^{(t)}\right) + b \tag{15}$$

where $\{(u^{(t)}, y^{(t)})\}_{t=1}^{N}$ is the training set and $y^{(t)} \in \{-1, +1\}$ represents the label. Given the histogram representation or high-level signature u of a test image, if $f(u) > 0$, then the image is assigned as a positive example; otherwise, it is assigned as a negative example. In the decision function, $\kappa(\cdot, \cdot)$ represents a reasonable Mercer kernel function that is used to measure the similarity between test images and training images.

In practice, the intersection kernel has been found to be effective on histogram representations [30]. For the image signature obtained from modified spatial pyramid pooling, it can be viewed as a special kind of histogram representation due to the local normalization. Let $u^{(x)}$ and $u^{(y)}$ denote the histogram representations obtained from the modified SPM or the high-level signatures obtained from the modified spatial pyramid pooling of two images. The intersection kernel is defined by the following formula:

$$\kappa\left(u^{(x)}, u^{(y)}\right) = \sum_{q=1}^{d_h} \min\left(u_q^{(x)}, u_q^{(y)}\right) \tag{16}$$

Large $\kappa(u^{(x)}, u^{(y)})$ indicates that the two images are extremely different from each other; otherwise, they are similar to each other. Thus, a binary classifier for two categories has been constructed. Multi-class classification can be performed with the guidance of the one-versus-one rule.

Table 1
Performance comparisons among various SPMs. Bold numbers represent the best results compared to others in this table.

Algorithm	Dimensionality	Scene-15 (%)	Sports-8 (%)
HOG + Original SPM	8400	81.84	81.92
HOG + SPM 1	8400	82.69	83.88
HOG + SPM 2	11,600	83.30	84.59
HOG + SPM 3	5200	**83.30**	**84.88**
SIFT + Original SPM	8400	81.40	81.46
SIFT + SPM 1	8400	82.47	82.96
SIFT + SPM 2	11,600	82.73	83.31
SIFT + SPM 3	5200	**82.82**	**83.42**

5. Experimental results and discussion

5.1. Datasets and setup

Our proposed algorithms are evaluated on the Scene-15 dataset [3] and the Sports-8 dataset [35]. The Scene-15 dataset has 15 categories of scenes, including coast, forest, mountain, open country, highways, inside city, tall building, street, bedroom, kitchen, living room, office, suburb, industrial and store. This dataset contains 4485 images, with the number of images in each category ranging from 200 to 400. Example images in each category of the 15-Scene dataset are shown in Fig. 9. The average resolution of the images is 300×250. The Sports-8 dataset contains eight sports events categories: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images) and rock climbing (194 images). Sample images are shown in Fig. 10. The size of these high-resolution images ranges from 800×600 to thousands of pixels per dimension.

Before extracting local visual descriptors, the maximum side (length/width) of the images is resized to 300 pixels with their original aspect ratios (For the Sports-8 dataset, we resize the maximum side to 400 pixels due to the high resolution of original images). The local visual descriptors are calculated from patches with 16×16 pixels that are densely sampled from each image in intervals of 8 pixels.

The classification of all experiments involving a multi-class SVM relies on LIBSVM using the intersection kernel. The multi-class SVM in LIBSVM [36] is performed with the one-versus-one rule, which is defined so that a classifier is learned to separate one class from another. A test sample is labelled as the class with the maximum number of votes.

5.2. Modified SPM based on HOG

In the modified SPM algorithm, we perform K-means clustering on a random subset of the training set to learn the visual words that correspond to the centres of K clusters. Then, each HOG descriptor is assigned as one visual word according to the closest centre. We count the occurrences of each visual word in different spatial regions to form the distribution histograms. Previous work showed that 400 is the proper number of visual words (i.e., the number of clusters) in [3].

We compare our modified SPM with several baseline algorithms in Table 1. To validate the effectiveness of our proposed new pyramid matching kernel and spatial partition scheme, the three types of spatial partition schemes in Fig. 11 combined with our new pyramid matching kernel are tested. A comparison of the original SPM and SPM 1 results shows that our proposed new pyramid matching kernel is superior to the original kernel. A comparison of the SPM 1 and SPM 3 results shows that our proposed spatial partition scheme further improves the recognition accuracy. This finding is consistent with the observation and assumption that the

Fig. 9. Example images of the Scene-15 dataset.

Fig. 10. Example images of the Sports-8 dataset.

horizontal and vertical partitions can provide more coherent and robust features from scene contents. Comparison of the SPM 2 and SPM 3 results shows that removing the finer partition seems to have no effect on the recognition accuracy, and it also decreases the dimensionality of the final feature vector and accelerates the computation.

In our modified SPM, an appropriate value of the kernel parameter a in Eq. (9) must be selected. To identify a reasonable value, the effect of this parameter on the performance is shown in Fig. 12. To perform a comprehensive assessment, three types of spatial partition schemes and two types of local visual descriptors are tested on the two datasets. The figures show that the recognition accuracy reaches the highest value when the kernel parameter is equal to 2, which implies that prior weights on higher levels are beneficial for scene recognition but large weights may have a negative effect on the performance.

Table 2
Performance comparison with other approaches for the Scene-15 dataset.

Method	Accuracy (%)
HOG + SPM 3 (Our modified SPM)	83.30
ScSPM [16]	80.28
Object Bank [8]	80.90
DSPM [19]	81.81
LAC [13]	82.18
DeepSCNet [39]	82.70
NDL [40]	82.75
CNF-SMN [41]	82.90
ASP [42]	83.10
DDSFL [37]	84.42

We also compare our results with the results of other prevalent approaches using only one type of local visual descriptor. As shown in Table 2, the recognition accuracy of our modified SPM on the Scene-15 dataset outperforms most of the benchmark works and is

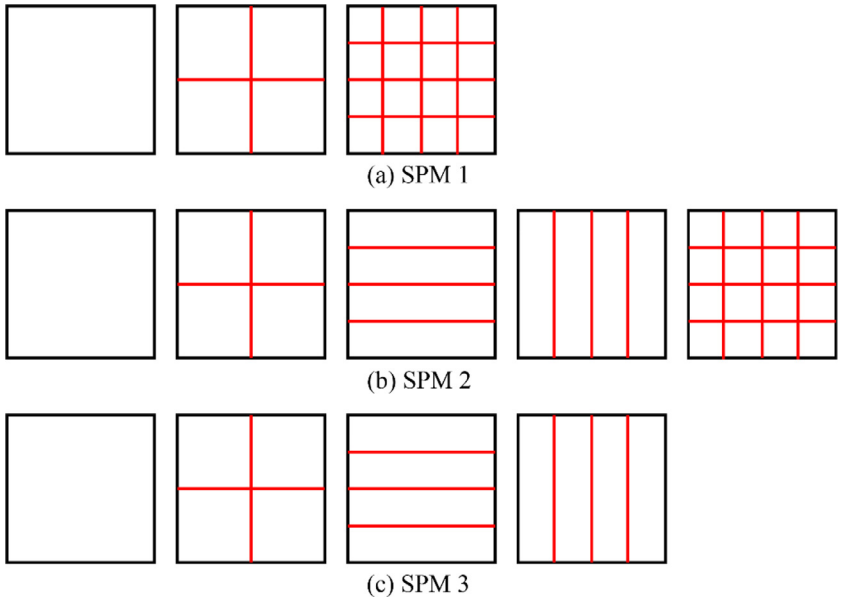

Fig. 11. Three types of spatial partition schemes. (a). SPM 1: standard spatial partition in the original SPM; (b). SPM 2; and (c). SPM 3: our proposed spatial partition scheme.

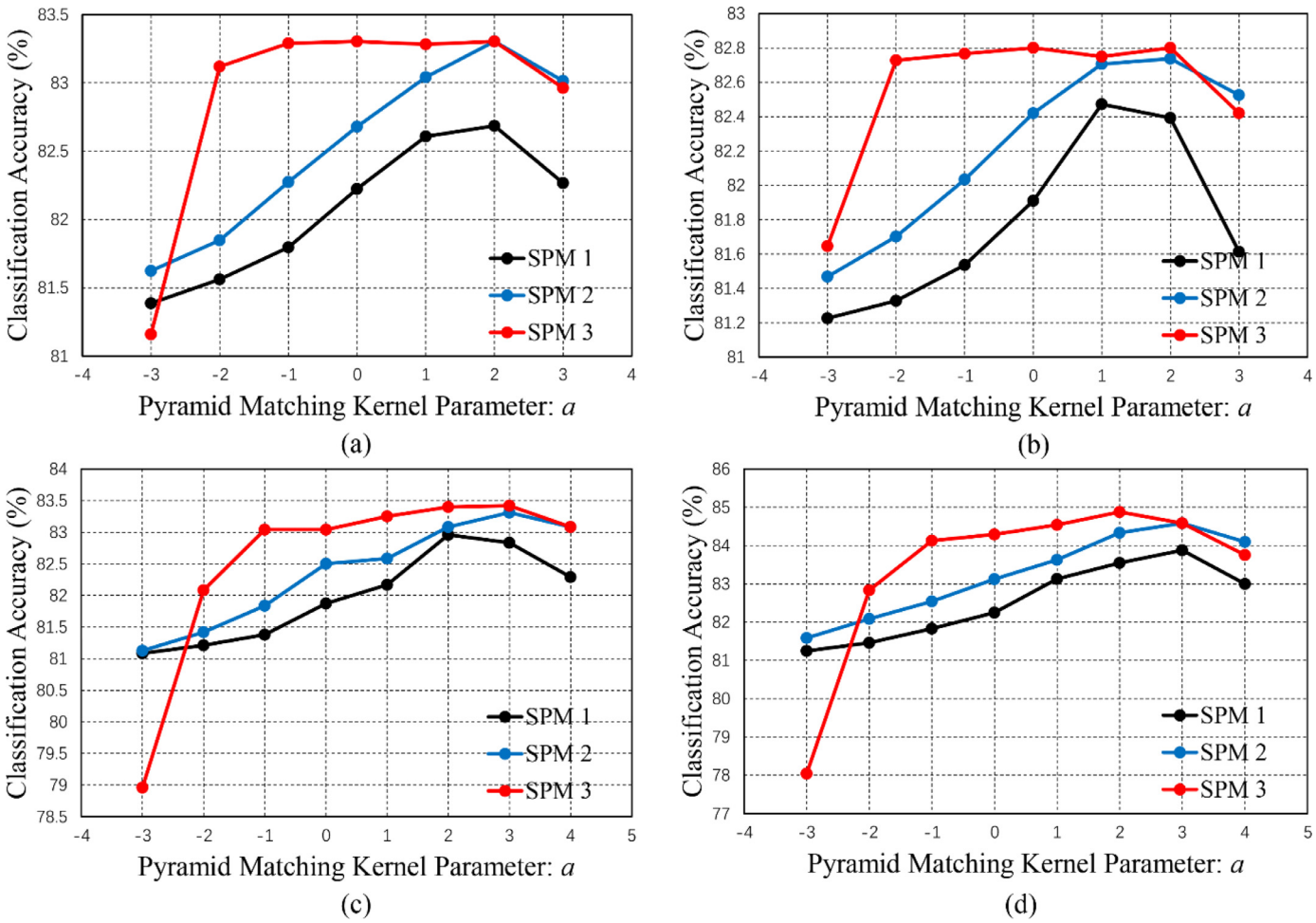

Fig. 12. Classification accuracy of various SPMs versus different pyramid matching kernel parameters using two types of local visual descriptors on two datasets. (a). HOG descriptors, Scene-15 dataset. (b). SIFT descriptors, Scene-15 dataset. (c). HOG descriptors, Sports-8 dataset. (d). SIFT descriptors, Sports-8 dataset.

Table 3
Performance comparison with other approaches for the Sports-8 dataset.

Method	Accuracy (%)
HOG + SPM 3 (Our modified SPM)	84.88
Object Bank [8]	76.30
RSP + Boosting [20]	79.60
ScSPM [16]	82.74
HIK + OCSVM [15]	83.54
DITC + SPM [38]	84.60
NDL [40]	84.62
DDSFL [37]	86.91

Table 4
Classification rate of modified spatial pyramid pooling based on various autoencoders. Bold numbers represent the best results compared to others in this table.

Algorithm	Number of hidden nodes	Scene-15 (%)	Sports-8 (%)
SPP-sparse autoencoder	512	77.10	76.32
	1024	78.58	**77.57**
	2048	**79.36**	77.15
SPP-denoising autoencoder	512	77.30	77.71
	1024	**78.49**	**79.59**
	2048	78.38	79.10
SPP-contractive autoencoder	512	77.91	77.71
	1024	77.94	**77.78**
	2048	**78.13**	77.15

slightly lower than that of the DDSFL [37]. Among these methods, ScSPM [16] constructed the spatial pyramid based on sparse coding and DSPM [19] formed the image features as a weighted sum of semi-local features over all pyramid levels; however, our modified SPM does not need the complex inference of sparse coding or partial least squares. Our modified SPM achieved consistent results for the Sports-8 dataset as presented in Table 3, and it outperformed the randomized partition pattern in RSP + Boosting [20] and the compact pyramid representation in DITC + SPM [38].

5.3. Modified spatial pyramid pooling based on various autoencoders

Before performing the modified spatial pyramid pooling, we attempt to use several autoencoder variants to encode local visual descriptors into mid-level features. The visible layer contains lower dimensions corresponding to the dimensionality of the local visual descriptor (i.e., 124 dimensions for HOG). The hidden layer has higher dimensions (e.g., 1024 dimensions for mid-level feature), and each unit represents a visual word. In these autoencoder variants, rectified linear functions are used for the hidden layer, which generates real zeros of activations and truly sparse representations [43,44]. In addition, the computations are less costly because of the absence of exponential functions. However, the hard saturation below the threshold of the rectified linear function produces difficulties for the reconstruction units because of the vanishing gradient problem. To avoid this, logistic sigmoid activation functions are used for the reconstruction layer. The parameter λ in the sparse autoencoder controls the sparsity of hidden representations (i.e., mid-level features) but excessive sparsity may reduce some useful components; therefore, it is selected by a grid search over the range from 10^{-3} to 10^{-9}. The standard deviation σ in the denoising autoencoder determines the magnitude of Gauss noise, which is searched in the range from 0.01 to 0.5. The parameter λ in the contractive autoencoder is selected from 10^{-3} to 10^{-5}.

Table 4 shows the experimental results using the modified spatial pyramid pooling based on three autoencoder variants. The encoding ability of the sparse autoencoder obviously prefers higher dimensionality and is sensitive to the variation of inputs (because the images in the Sports-8 dataset have higher resolutions). The mid-level features transformed form the denoising autoencoder and the contractive autoencoder appear to be more robust, and different dimensionalities have little influence on the mid-level features. However, the low classification rates suggest that the mid-level features learned from these autoencoders are insufficient for scene recognition.

Table 5
Classification rate of modified spatial pyramid pooling based on new sparse autoencoder.

Algorithm	Number of hidden nodes	Scene-15 (%)	Sports-8 (%)
SPP-modified sparse Autoencoder (HIK SVM)	1024	79.62	77.40
	2048	79.86	76.98
SPP-modified sparse Autoencoder (Linear SVM)	1024	80.85	81.31
	2048	81.97	81.36

Table 6
Running time and memory requirements of the two proposed algorithms.

Algorithm	Modified SPM	SPP-modified sparse Autoencoder (Linear SVM)
Main computational complexity	$\mathcal{O}(n \cdot m \cdot d_x) + \mathcal{O}(N)$	$\mathcal{O}(n \cdot d_x \cdot d_h) + \mathcal{O}(1)$
Memory complexity	$\mathcal{O}(m \cdot d_x + N \cdot d_l)$	$\mathcal{O}(d_h \cdot (d_x + 1))$
Testing time (s)	0.06	0.05

To improve the recognition performance, the modified sparse autoencoder proposed in Section 4.2.2 is applied to the modified spatial pyramid pooling algorithm, the experimental results are shown in Table 5. It is worth noting that the newly learned mid-level features prefer linear SVM rather than the intersection histogram kernel SVM, which accelerates the computation. Meanwhile, the improvement of the recognition accuracy validates the effectiveness of the modified sparse autoencoder.

5.4. Running time and memory requirements

All experiments are performed on a CPU server with a Xeon E5-2630 2.4 GHz CPU with 8 cores. The time for extracting local visual descriptors is proportional to the input image size, for example, the average time per image of the Scene-15 dataset is 0.09 s, this time will not be considered in the following running time. As the training phase of the two proposed algorithms is offline, here we mainly discuss the time performance and memory management of the test phase, which have been listed in Table 6.

In the modified spatial pyramid matching, the main task is to assign the nearest visual word to each local descriptor, the theoretical computational complexity of this step is $\mathcal{O}(n \cdot m \cdot d_x)$, where n is the number of local visual descriptors from an image, m is the number of clusters and d_x is the dimensionality of local visual descriptors (In our previous experiments, $m = 400$, $d_x = 124$). The histogram intersection kernel SVM has a computational complexity $\mathcal{O}(N)$, where N is the training size. The average time for testing each image is 0.06 s. The modified spatial pyramid matching needs a memory complexity $\mathcal{O}(m \cdot d_x + N \cdot d_l)$ to store the visual words and the distribution histograms of training samples, where d_l is the dimensionality of the distribution histogram.

In the modified spatial pyramid pooling based on the modified sparse autoencoder, the most expensive steps is to calculate the mid-level features, which requires a computational complexity $\mathcal{O}(n \cdot d_x \cdot d_h)$, where d_h is the dimensionality of hidden representations (In our previous experiments, $d_x = 124$, $d_h = 1024$ or 2048), it seems to be greater, however, the linear SVM only requires a constant complexity. The average time for testing each image is 0.05 s when $d_h = 1024$. Moreover, it needs a memory complexity $\mathcal{O}(d_h \cdot (d_x + 1))$ to store the weight matrix W_1 and bias vector b_h, which is much smaller than the memory requirements of the mod-

ified spatial pyramid matching because $N \cdot d_l$ is much larger than $d_h \cdot (d_x + 1)$.

6. Conclusions

In this paper, we present two methods for scene recognition. In the modified SPM method, the HOG descriptors are regarded as low-level features. Our proposed new pyramid matching kernel and spatial partition scheme achieved better performance than the original SPM, using lower feature dimensionality. In the modified spatial pyramid pooling method based on various autoencoders, various constraints are added into the basic autoencoder to learn interesting and useful representations. This method is different from other autoencoder models that extract features from pixels. Different properties of mid-level features are considered, such as sparsity, robustness and contraction. The experimental results indicated that the mid-level features learned from various autoencoders have limited power for scene recognition. Some explorations on the sparse autoencoder have been performed to improve the recognition accuracy. In the future work, some other constraints or supervised learning may be introduced in the autoencoder to obtain better mid-level features for recognition.

Conflict of interest

None declared.

Acknowledgement

This research is partially supported by The Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and also partially supported by JSPS KAKENHI Grant Number 15K00159.

References

[1] J. Xiao, J. Hays, K.A. Ehinger, A. Torralba, SUN database: large-scale scene recognition from abbey to zoo, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2010, pp. 3485–3492.
[2] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, Towar. Categ. Object Recognit. (2003) 1470–1477.
[3] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2006, pp. 2169–2178.
[4] A. Oliva, A. Torralba, Building the gist of a scene: the role of global image features in recognition, Progr. Brain Res. 155 (2006) 23–36.
[5] J. Wu, J.M. Rehg, Centrist: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.
[6] X. Meng, Z. Wang, L. Wu, Building global image features for scene recognition, Pattern Recognit. 45 (1) (2012) 373–380.
[7] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognit. 46 (2) (2013) 483–496.
[8] L. Li, H. Su, E.P. Xing, L. Fei-fei, Object bank: a high-level image representation for scene classification & semantic feature sparsification, Adv. Neural Inf. Process. Syst. (2010) 1–9.
[9] L. Torresani, M. Szummer, A. Fitzgibbon, Efficient object category recognition using classemes, in: Proc. European Conf. Comput. Vis., 2010, pp. 776–789.
[10] X. Zhou, X. Zhuang, H. Tang, M.H. Johnson, T.S. Huang, Novel Gaussianized vector representation for improved natural scene categorization, Pattern Recognit. Lett. 31 (8) (2010) 702–708.
[11] J. Qin, N.H.C. Yung, Scene categorization via contextual visual words, Pattern Recognit. 43 (5) (2010) 1874–1888.
[12] L. Zhou, Z. Zhou, D. Hu, Scene classification using a multi-resolution bag-of-features model, Pattern Recognit. 46 (1) (2013) 424–433.
[13] K. Hotta, Local autocorrelation of similarities with subspaces for shift invariant scene classification, Pattern Recognit 44 (4) (2011) 794–799.
[14] K. Hotta, Local co-occurrence features in subspace obtained by KPCA of local blob visual words for scene classification, Pattern Recognit. 45 (10) (2012) 3687–3694.
[15] J. Wu, J.M. Rehg, Beyond the Euclidean distance: creating effective visual codebooks using the histogram intersection kernel, in: Proc. IEEE Int. Conf. Comput. Vis, 2009, pp. 630–637.
[16] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2009, pp. 1794–1801.
[17] S. Gao, I.W.H. Tsang, L.T. Chia, P. Zhao, Local features are not lonely -Laplacian sparse coding for image classification, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2010, pp. 3555–3561.
[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Locality-constrained linear coding for image classification, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2010.
[19] T. Harada, Y. Ushiku, Y. Yamashita, Y. Kuniyoshi, Discriminative spatial pyramid, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2011, pp. 1617–1624.
[20] Y. Jiang, J. Yuan, G. Yu, Randomized spatial partition for scene recognition, in: Proc. European Conf. Comput. Vis., 2012, pp. 730–743.
[21] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, 2005, pp. 524–531.
[22] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2005, pp. 886–893.
[23] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[24] K. Graumanand, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, in: Proc. IEEE Int. Conf. Comput. Vis., 2005, pp. 1458–1465.
[25] P. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples without local minima, Neural Netw. 2 (1989) 53–58.
[26] N. Japkowicz, Jos, M.a Gluck, Nonlinear autoassociation is not equivalent to PCA, Neural Comput. 12 (2000) 531–545.
[27] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, J. Mach. Learn. Res. 9 (2010) 249–256.
[28] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1096–1103.
[29] S. Rifai, X. Muller, Contractive auto-encoders: explicit invariance during feature extraction, in: Proc. 28th Int. Conf. Mach. Learn., vol. 85, 2011, pp. 833–840.
[30] Y. Boureau, F. Bach, Learning mid-level features for recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2010, pp. 2559–2566.
[31] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Int. Conf. Neural Info. Proc. Sys, 2012, pp. 1097–1105.
[32] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[33] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proc. IEEE Int. Conf. Acoust. Speech Sig. Proc., 2013, pp. 6645–6649.
[34] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: Int. Conf. Learn. Represent., 2015, pp. 1–15.
[35] L.J. Li, L. Fei-Fei, What, where and who? classifying events by scene and object recognition, in: Proc. IEEE Int. Conf. Comput. Vis., 2007, pp. 1–8.
[36] C. Chang, C. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2013) 1–39.
[37] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, Exemplar based deep discriminative and shareable feature learning for scene image classification, Pattern Recognit. 48 (10) (2015) 3004–3015.
[38] N.M. Elfiky, F.S. Khan, J.V.D. Weijer, J. Gonzalez, Discriminative compact pyramids for object and scene recognition, Pattern Recognit. 45 (4) (2012) 1627–1636.
[39] S. Zhang, J. Wang, X. Tao, Y. Gong, N. Zheng, Constructing deep sparse coding network for image classification, Pattern Recognit. 64 (2017) 130–140.
[40] J. Hu, Y. Tan, Nonlinear dictionary learning with application to image classification, Pattern Recognit. (2017).
[41] X. Song, S. Jiang, L. Herranz, Y. Kong, K. Zheng, Category co-occurrence modeling for large scale scene recognition, Pattern Recognit. 59 (2016) 98–111.
[42] Y. Liu, Y. Zhang, X. Zhang, C. Liu, Adaptive spatial pooling for image classification, Pattern Recognit. 55 (2016) 58–67.
[43] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: AISTATS Proc. 14th Int. Conf. Artif. Intell. Stat., vol. 15, 2011, pp. 315–323.
[44] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. 30th Int. Conf. Mach. Learn., vol. 28, 2013, p. 6.

Lin Xie is currently a master student at School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology. His current research interests are in the areas of scene recognition and machine learning.

Feifei Lee received her Ph.D. degree in electronic engineering from Tohoku University in Japan, in 2007. She is currently a professor at the University of Shanghai for Science and Technology. Her research interests include pattern recognition, video indexing, and image processing.

Li Liu received the Ph.D. degree in pattern recognition and intelligent system from East China Normal University, Shanghai, China, in 2015. She was with the Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, QC, Canada, from 2013 to 2014 as a visiting doctoral student, and in 2016 as a visiting scholar. She is currently a lecturer with Nanchang University. Her research interests include pattern recognition, computer vision, and document image analysis.

Zhong Yin received the Ph.D. degree in control science and engineering from the East China University of Science and Technology. He has been a lecturer at School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China, since 2015. His research interests include intelligent human-machine systems, biomedical signal processing and pattern recognition.

Yan Yan is currently pursuing the Ph.D. degree in control science and engineering at University of Shanghai for Science and Technology. Her current research interests are in the areas of image recognition, computer vision and machine learning.

Weidong Wang is currently a master student at School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology. His current research interests are in the areas of pattern recognition and image retrieval.

Junjie Zhao is currently a master student at School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology. His current research interests are in the areas of pattern recognition and video retrieval.

Qiu Chen received Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2004. Since then, he has been an associate professor at Tohoku University and Kogakuin Univeristy. His research interests include pattern recognition, computer vision, information retrieval and their applications. He is also a guest professor at the University of Shanghai for Science and Technology. Dr. Chen serves on the editorial boards of several journals, and he is a member of IEEE.